
COMPUTER SCIENCE

For Rwandan Schools

Senior 4

Teacher's Book

Contents

Introduction	1
Basic information	1
Pedagogical approaches	14
Planning for teaching.....	22
Assessment.....	27
Content Map.....	31
Unit 1: Computer Fundamental	48
Unit 2: Computer Architecture, Assembling and disassembling Computer.....	57
Unit 3: Safe and Ethical Use of Computers	71
Unit 4: Computer Software Installation.....	79
Unit 5: Number Systems	86
Unit 6: Boolean Algebra and Logic Gates.....	101
Unit 7: Introduction to Computer Algorithm.....	110
Unit 8: Control Structures and Arrays.....	121
Unit 9: Introduction to Computer Programming.....	131
Unit 10: Introduction to C++ Programming.....	140
Unit 11: Operators and Expressions C++ Programming.....	151
Unit 12: Control Statements in C++	159
Unit 13: Functions in C++ Programming.....	168
Unit 14: Arrays in C++ Programming.....	177
Unit 15: Introduction to Operating Systems	184
Unit 16: HTML–Based WEB Development.....	193
Unit 17: Cascading Style Sheets	203

INTRODUCTION

Section I Basic information

I.1 Organisation of the book

This teacher's guide is organised into two main sections. Part 1 is the general introduction section detailing pedagogical issues. Part 2 is the main topics area. It gives the details of the expected learning units as organised in the learner's book. The main elements of Part 2 are:

- **Topic Area** page - detailing the various **Sub-topic Areas** and the units covered under the topic area.
- **Unit heading** – this is accompanied by some text in the Student's Book to motivate the learners. Also, the total number of periods per unit is given.
- **Key Unit Competence:** This is the competence, which will be achieved once students have met all the learning objectives in the unit.
- **Outline of main sections** in the unit – is a quick summary of the subtopics covered under the unit.
- **Learning Objectives:** The content in this area is broken down into three categories of learning objectives, that is, knowledge and understanding; skills; attitudes and values.
- **Knowledge and understanding:** As in the existing curriculum, knowledge and understanding is very important.
- **Skills:** It is through the skills that students apply their learning and engage in higher order thinking. These skills relate to the upper levels of

Bloom's taxonomy and they lead to deep rather than surface learning.

Attitudes and values: This captures appropriate attitudes and values that relate to the unit.

Links to other subjects: It is important for learners to gain an understanding of the interconnections between different subjects so that learning in each subject is reinforced across the curriculum. This platform does exactly that. It prepares the teacher to pass this information to the learners so that they are aware!

Assessment Criteria: This is meant to evaluate whether learners have succeeded in achieving the Key Unit Competence(s) intended. This section will help the teacher in assessing whether the unit objectives have been met.

Background information: This is the introduction part of the unit. It aims at giving insights to the teacher on the subject matter.

Additional information for the teacher: This section gives more information than what the syllabus recommends for purposes of preparing the teacher to answer tough questions from learners.

Learning Activities: These are given per lesson and have these sub-sections:

- Lesson titles
- Specific objectives of the lesson
- Materials and learning resources
- Teaching methodology
- Suggested teaching/learning approach
- Generic competencies covered
- Cross-cutting issues covered
- Special needs and multi-ability learning
- Formative assessment
- Extended exercises/activities for fast learners and remedial (reinforcement) exercises/activities for slow learners.
- Answers to self-evaluation exercises

These are repeated across all periods until the end of the unit followed by the answers or tips on the test your competence questions at the end of every unit.

1.2 The Structure of the syllabus

Computer science is a competence based syllabus for combinations subject is taught and learned at Advanced Level. The syllabus is structured in **Topic Areas**, and then further broken down into **Units**. The units have the following elements:

- Number of Periods required to cover each topic.
- Key Unit Competency whose achievement is pursued by all teaching and learning activities undertaken by both the teacher and the learners. The Key Competency is broken into three types of learning

objectives as follows:

- *Type I:* Learning Objectives relating to Knowledge and Understanding. These are associated with Lower Order Thinking Skills or **LOTS**.
 - *Type II and Type III:* These Learning Objectives relate to acquisition of skills, Attitudes and Values. They are associated with Higher Order Thinking Skills or **HOTS**.
 - These Learning Objectives are actually considered to be the ones targeted by the present reviewed syllabus.
- **Content** area which indicates the scope of coverage of what a teacher should teach and learner should learn in line with stated objectives, skills and attitude.
 - Suggested teaching and learning activities that are expected to engage learners in an interactive learning process as much as possible (learner-centered and participatory approach).
 - Link to other subjects, assessment criteria and materials (or resources) that are expected to be used in teaching and learning process.

The Computer Science syllabus for Senior 4 has 17 units. These units have been covered by the following units in the student's book:

1. Computer Fundamentals
2. Computer Architecture and Assembly

3. Safe and ethical use of computers.
4. Computer Software Installation
5. Number Systems
6. Boolean Algebra and Logic Gates
7. Introduction to Computer Algorithm
8. Control structures and one dimension array
9. Introduction to Computer Programming
10. Introduction to C++ programming
11. Operators and Expression in C++
12. Control Statements in C++
13. Function in C++ Programming
14. Arrays in C++ Programming
15. Introduction to Operating Systems
16. HTML-Based Web Development
17. Cascading Style Sheet

1.3 Background Information on the new curriculum

The goal of computer science is to develop a competence-based society that fits in the globalization process characterised by knowledge economy and competition. Competence-based learning refers to systems of instruction, assessment, grading, and academic reporting that are based on students demonstrating that they have acquired and learned the prerequisite knowledge, skills and attitudes as they progress through their education.

This is the motivation behind paradigm shift from knowledge-based to competence-based curriculum that resulted to comprehensive review of

the national curriculum in Rwandan education system. As part of the changes in the curriculum, Computer Science syllabus was reviewed to ensure that the syllabus is responsive to the needs of the learner with a shift from knowledge-based learning to competence-based learning. Another reason for curriculum review was inclusion of the computer science component leading to the new subject named “Computer Science Competence Based Syllabus.”

Apart from being integrative, the new syllabus guides the interaction between the teacher and the learner in the learning process. It further puts greater emphasis on skills a learner should acquire during each unit of learning. As a competency-based syllabus, it elaborates on the competence-based aspects of **knowledge, skills** and **attitudes** and values in computer science.

The competence based curriculum is in line with various government policies and aspirations as stipulated in various policy documents including The Education Sector Strategic Plan 2013 and Vision 2020 that emphasis quality education and put a lot of emphasis on science, technology and innovation; Poverty Reduction Strategy (EDPRSII) and the draft ICT Policy in Education 2014 that stresses the need to avail ICT to all learners. The need to address all these aspirations necessitated a comprehensive review of the Lower Secondary Curriculum

1.4 Rationale of teaching and learning Computer Science

Computer science is a powerful subject of modern life which has changed every aspect to the human society. With computer science, different computing devices have been developed and they are helping in daily life. It is important to highlight inventions of computers, embedded systems, telephony for communication, and automation of different human activities through different applications facilities, usage of computing technologies in medicines, education, finance, socio-economic, military, security, spatial sciences, and weather forecast.

In Rwandan society Computing technologies drive the country development through communication, corporate growth, economic development, financial transaction, job creation, foster the value of investments, living standards, patterns of work and leisure.

11.5 Types of Competences and their acquisition

The national policy documents, based on the national aspirations, identify '**Basic Competences**' alongside the '**Generic Competences**' that will develop higher order thinking skills.

a) **Basic competences**

Basic competences are addressed in the broad subject competences and in objectives include:

i) **Literacy**

- Reading a variety of texts accurately and quickly.
- Expressing ideas, messages and events through writing legible texts in good hand-writing with correctly spelt words.
- Communicating ideas effectively through speaking using correct phonetics of words.
- Listening carefully for understanding and seeking clarification when necessary

ii) **Numeracy**

- Computing and converting numbers in the four base systems: decimal, binary, octal and hexadecimal.
- Using algorithms and computer programs to manipulate numbers, symbols, quantities, computer memory, and other computing tasks.
- Using computer programs to solve problems related to everyday activities like commercial and financial management.
- Interpreting basic statistical data using programs, tables, diagrams and charts.

b) **Generic competences**

Generic competences must be emphasized and reflected in the teaching and learning of computer science. The national policy documents, based on the national aspirations, identify 'Basic

Competences' alongside the 'Generic Competences' that will develop higher order thinking skills. Basic Competencies are addressed in the stated broad subject competences and in the objectives highlighted on year basis and in each of the units of learning.

The selection of types of learning activities must focus on what competencies the learners are able to demonstrate throughout and at the end of the learning process. Basic competencies must be emphasised during teaching and learning process.

They are briefly described below and teachers must ensure that learners are engaged in tasks that help them to acquire the competences.

1. Critical thinking and problem solving skills: The acquisition of these skills will help learners think imaginatively and broadly to evaluate and find solutions to problems encountered in all situations.

2. Creativity and innovation: The acquisition of these skills will help learners take initiative and use imagination beyond the knowledge provided to generate new ideas and construct new concepts.

3. Research skills: This will help learners find answers to questions based on existing information and concepts and to explain phenomena based on findings from information gathered.

4. Communication in official languages: Teachers, irrespective of

not being teachers of language, will ensure the proper use of the language of instruction by learners which will help them to communicate clearly and confidently. It will also assist learners to convey ideas effectively through speaking and writing and by using the correct language structure and relevant vocabulary.

5. Cooperation, inter personal management and life skills: This will help the learner to cooperate with others as a team in whatever task are assigned and to practice positive ethical moral values and respect for the rights, feelings and views of others. Learners will also perform practical activities related to environmental conservation and protection. These skills will also assist learners to advocate for personal, family and community health, hygiene and nutrition and respond creatively to the variety of challenges encountered in life.

6. Lifelong learning: The acquisition of this skill will help learners to update their knowledge and skills with minimum external support and to cope with the evolution of knowledge advances for both personal fulfillment and in areas that need improvement and development.

1.6 Computer Science and developing competences

Computer Science contributes to a learner's development of critical thinking, research and problem solving, creativity

and innovation, communication and cooperation skills. This will be achieved by conceptualising, analysing, synthesising, evaluating, manipulating and applying various programming techniques, database management and networking to appropriate contexts of real life

Through computer science, learners will identify problems and conduct analysis by considering relevant functional, practical, human and socioeconomic factors. They will also draw up specifications for the computer-based solutions to problems from a range of suitable resources.

Computer science will also contribute to promotion of self-reliance of Rwandan through provision of necessary knowledge, skills, attitude and values.

Furthermore, computing solutions will enhance nation's integrity; peace and respect of others which are key values for development of the country.

1.6 Important Attitudes in Computer Science

1.6.1 Attitude in learners

There are certain useful attitudes, which the teacher should help to develop in the learners as they carry out investigations in computer science. Computer science as a problem solving discipline is expected to make an impact on a learner's general behaviour.

The nature of computer science method demands learners to be honest with themselves as they record results and make unbiased conclusions. They should also be aware of the danger involved in

making generalizations out of limited information. They should be open-minded and able to distinguish between propaganda and truth.

Some of the Computer Science attitudes that learners should develop include:

- **Practical approach** – to problem solving. Learners should seek answers to their questions and problems by carrying out investigations wherever possible.
- **Responsibility** – A learner should be responsible enough to effect tasks apportioned and take good care of apparatus during and after an investigation.
- **Cooperation** – Learners will often be working in groups while carrying out investigations and need therefore to cooperate with all other members of the group.
- **Curiosity** – Learners should have a curious attitude as they observe things and events around them. This is the first step towards solving a problem.
- **Self-confidence** – Learners should have the will to attempt to solve a problem. The feeling of self-confidence can be strengthened in young learners if they experience many small successes that win approval and encouragement from the teacher. The problems which learners attempt to solve should not be so difficult that they lead to frustration.
- **Honesty** – As they make observations, record, analyse results

and draw conclusions.

- **Patience** – Learners should be patient for the results of an experiment which may take time to manifest.

1.6.2 Attitude in teachers

- Engage students in variety of learning activities
- Apply appropriate teaching and assessment methods
- Adjust instructions to the level of the learner
- Creativity and innovation
- Makes connections/relations with other subjects
- Show a high level of knowledge of the content
- Develop effective discipline skills manage adequately the classroom
- Good communicator
- Guide and counsellor
- Passion for children teaching and learning.

1.7 Broad Computer Science Competences

Broad competences that must be emphasized and reflected in the teaching and learning process are highlighted below:

- Use computational competences to contribute significantly to GDP growth by improving government services, business efficiency and productivity for long term and sustainable economic competitiveness.
- Develop life skills, practical and entrepreneurial skills
- Apply computational thinking, logical and algorithmic precision in problem solving and creativity.
- Do maintenance of computer hardware and perform installation/uninstallation of software
- Use computer ergonomically to avoid health related risk
- Respect ethical issues related to piracy, software licensing, copyright, individual privacy
- Apply the stages of the software development life cycle (i.e., problem definition, analysis, design, testing, implementation, maintenance)
- Identify errors in a program and apply principles of debugging
- Analyze cultural, legal, and ethical issues and responsibilities of digital citizens, organizations, and government entities (e.g. privacy issues related to internet use, data protection).
- Analyze issues related to malicious software, social engineering, and security awareness.
- Design small web-based, desktop and mobile applications
- Design, install, maintain and administer a database
- Install and administer Windows and Linux kernel-based operating systems
- Maintain common technological tools
- Design, maintain and administer

computer networks

- Create and capture images, audio, videos; edit them using appropriate software
- Show the following skills in his/her everyday life: communication, research, practical problem solving, observation, creativity and innovative skills.

1.8 Key competences at the end of Secondary 4

- Explain the computer system, evolution, role, architecture and classification
- Identify different numbers base systems and perform conversion between them
- Identify different logic gates, theorems of Boolean algebra and evaluate Boolean expressions
- Utilize the laws of Boolean algebra on Boolean expressions and draw a simple logic circuit using logic gates
- Integrate safety guidelines, ergonomics and ethical issues to have a good working environment
- Disassemble and assemble computer; do minor maintenance
- Identify different computer ports and connectors
- Derive a suitable algorithm for a computational problem using variables, expressions, reading and writing functions, and loops
- Describe programming paradigms
- Transform an algorithm in C ++ and apply functions, arrays and control

statements in C++ program

- Explain the Evolution, role, types and function of the Operating System
- Build standards compliant web pages using XHTML and CSS

1.9 Cross-cutting issues to be infused during learning

These are emerging issues which need to be incorporated in the learning process. Each of the cross-cutting issues has its own important programme of learning reflecting key national priorities. This learning is integrated into the syllabuses of subjects across the curriculum rather than each issue having a dedicated timetable slot of its own. As a result of this integration, the learning activities in the units of subjects across the curriculum incorporate all the learning associated with the cross-cutting issues. The eight cross-cutting issues are:

a) Peace and Values Education

The need for Peace and Values Education in the curriculum is obvious. Peace is clearly critical for society to flourish and for every individual to focus on personal achievement and their contribution to the success of the nation. Values education forms a key element of the strategy for ensuring young people recognise the importance of contributing to society, working for peace and harmony and being committed to avoiding conflict.

b) Financial Education

Financial education makes a strong

contribution to the wider aims of education. It makes learning relevant to real life situations. It aims at a comprehensive financial education program as a precondition for achieving financial inclusion target and improves the financial capability of Rwandans. Financial education has a key role of not only improving knowledge of personal but also transforming this knowledge into action. It provides the tools for sound money management practices on earnings, spending, saving, borrowing and investing. Financial education enables people to take appropriate financial services both formal and informal that are available to them and encourages financial behaviours that enhance their overall economic well-being.

c) *Standardisation Culture*

Standardisation Culture develops learners' understanding of the importance of standards as a pillar of economic development and in the practices, activities and lifestyle of the citizens. It is intended that the adoption of standardization culture should have an impact upon health improvement, economic growth, industrialization, trade and general welfare of the people. While education is the foundation and strength of our nation, standards are one of the key pillars of sustainable economic development.

d) *Genocide Studies*

Genocide Studies provides young people with an understanding of the

circumstances leading to the genocide and the remarkable story of recovery and re-establishing national unity. Genocide Studies helps learners to comprehend the role of every individual in ensuring nothing of the sort ever happens again.

The intent of a cross-cutting curriculum around the topic of genocide is to fight against genocide, genocide denial, and genocide ideology; and to equip students with a more fundamental and comprehensive understanding of the genocide, thereby preventing further human rights violations in the future and enabling Rwanda's population of young people to more competently and thoughtfully enter the workforce. So, it needs to be emphasized.

e) *Environment and sustainability*

The growing awareness of the impact of the human race on the environment has led to recognition of the need to ensure our young people understand the importance of sustainability as they grow up and become responsible for the world around them. One of the rising threats to the environment is the poor management of e-waste. Learners need skills and attitudes that will enable them in their everyday life to conserve the environment and address climate change issue and to have a sustainable livelihood.

f) *Gender education*

There is a strong moral imperative to afford every individual their basic human rights. Gender inequality results in women and girls being treated less

favourably than men. A strongly negative impact of unequal treatment, which affects the nation as a whole, is the fact that it results in women being held back and their talents and abilities not being fully realised. With a good understanding of the principles of Gender Equality, it is intended that future generations will ensure that the potential of the whole population is realised.

g) *Comprehensive sexuality education (HIV/AIDS, STI, Family planning, Gender equality and reproductive health)*

Comprehensive sexuality education, which is age appropriate, gender sensitive and life skills based can provide young people with the knowledge and skills to make informed decisions about their sexuality and life style. Preparing children and young people for the transition to adulthood has been one of humanity's greatest challenges with human sexuality and relationships at its core. Few young people receive adequate preparations for their sexual lives. This leaves them potentially vulnerable to coercion, abuse and exploitation. Unintended pregnancy and sexually transmitted infections (STIs) including HIV/AIDS. Many young people approach adulthood faced with conflicting and confusing messages about sexuality and gender. This is often exacerbated by embarrassment, silence, disapproval and open discussion of sexual matters by adults (parents, teachers) at very time when it is most needed.

Comprehensive sexuality education

supports a rights- based approach in which values such as respect, acceptance tolerance, equality, empathy and reciprocity are inextricably linked to universally agreed human rights. A clear message concerning these dangers and how they can be avoided, from right across the curriculum, is the best way to ensure that young people understand the risks and know how to stay healthy.

h) *Inclusive Education*

Inclusive education involves ensuring all learners are engaged in education and that they are welcomed by other students so that everyone can achieve his/her potential. Inclusive practice embraces every individual regardless of gender or ability including those with learning difficulties and disabilities. The almost focus of inclusive curriculum is on ensuring participation in education of learners with different learning styles and other difficulties. To be successful, it entails a range of issues including teacher's positive attitudes, adapting the learning resources, differentiation of teaching and learning methods and working together. Overall, the benefits of an inclusive curriculum extend to all learners.

1.7 Special needs education and inclusivity

All Rwandans have the right to access education regardless of their different needs. The underpinnings of this provision would naturally hold that all

citizens benefit from the same menu of educational programs. The possibility of this assumption is the focus of special needs education. The critical issue is that we have persons/ learners who are totally different in their ways of living and learning as opposed to the majority. The difference can either be emotional, physical, sensory and intellectual learning challenges traditionally known as mental retardation. These learners equally have the right to benefit from the free and compulsory basic education in the nearby ordinary/mainstream schools. Therefore, the schools' role is to enrol them and also set strategies to provide relevant education to them. The teacher therefore is requested to consider each learner's needs during teaching and learning process. Assessment strategies and conditions should also be standardised to the needs of these learners. Also, ensure that you include learners with special educational needs in classroom activities as much as possible.

The special needs children can fall in any of the following common categories:

- Physical difficulties
- Visual difficulties
- Hearing difficulties
- Mental difficulties
- Genocide traumatized learners

The teacher should identify such cases and help facilitate the affected learners learning. For example, learner's with visual and hearing difficulties should sit near the teacher's table for easy

supervision and assistance. The following are some suggestions on how to support special needs children in your class.

(a) Learners with physical difficulties

In this group of learners, the affected areas are normally some body parts, especially the limbs. There may be partial or total loss of use of the limbs. In case the legs are affected, the learners will need assistance during activities that involve movement. This could be during a nature walk and other activities that learners have to stand for some reason. The teacher should organize for the learner's ease of movement around. The learner should also be given time to catch up with the others.

In case the hands are affected, the learners should be given more time to finish their work. In both cases, the learners should not be pressurized to do things that can cause injury or ridicule.

(b) Learners with visual difficulties

These learners normally have problems with their eyesight. They should sit in a position where they are able to see the chalkboard without straining

Note: The learner could be longsighted or short sighted.

The material to be observed should be brought closer to the learner and a magnifying lens used where necessary. The teacher should use large diagrams, charts and labels. In some cases, the learners can be allowed to touch and feel whatever they are looking at. Other learners can assist by reading aloud. The

lighting system in the classroom can also be improved.

The teacher should read aloud most of the things he/she writes on the chalkboard.

(c) Learners with hearing difficulties

The affected part in this case is the ear. The learner should have hearing aids. The teacher should use as many visual aids as possible. They should also project their voice and always talk while facing the learners. Use of gestures and signs while talking helps the learner figure out what the teacher is saying as well.

(d) Learners with speech difficulties

A common example in a normal class is the **stammerer**. They always speak with a lot of difficulties. The teacher should be patient with them and encourage such learners to express themselves in their own way. Such learners should be given more written exercises.

(e) Learners with mental difficulties

The teacher should try to identify the nature and level of the mental difficulty. Learners with mental difficulties should then be given special assistance and attention at an individual level. They can be given special tests or assessments. In general, all the learners with difficulties

should be reinforced promptly. This encourages and motivates them.

The teacher and the rest of the class should never ridicule learners with any of the difficulties. Note that generally, people with any kind of disability can be very sensitive to any kind of negative comments or criticism.

Remind them that ‘Disability is not inability’.

The teacher should avoid giving privileges where the learners do not deserve them. Treat them fairly but not with undue favours. In extreme cases it can be recommended for the learners to join a special school.

(f) Genocide traumatised learners

Studies have shown that learners from families that were affected by genocide suffer post-traumatic stress disorder (PTSD). As such, they need to be treated as a special case. As a teacher, you need to be careful when dealing with such learners. Also, the teacher needs to be in control especially when the topic under discussion touches on genocide issues. Any language that may elicit emotional reactions from learners either by fellow learners or by the teacher him or herself should be avoided.

2.1: Methods of teaching

There are a variety of methods through which a teacher can help the pupils to learn. These include :

- (a) Lecture method
- (b) Discovery or practical activity
- (c) Group, class or pair discussion
- (d) Project method
- (e) Educational visit/ field trips
- (f) Teacher demonstration

The choice of a teaching method is influenced by:

- The skills, attitudes and knowledge to be learned.
- Learning and teaching aids available
- The academic ability of the the learners in the class.
- The classroom environment
- The teacher's personal preference
- The requirements of the Computer Science syllabus

(a) Lecture method

This is the traditional way of teaching whereby the teacher explains something while the learners listen. After the teacher has finished, the learners may ask questions. However, in delivering a competence-based curriculum, is highly discouraged.

(b) Guided Discovery

In this technique, the teacher encourages learners to find out answers to problems by themselves. The teacher does this by:

- Giving learners specific tasks to do
- Giving learners materials to work with
- Asking structured or guided

questions that lead learners to the desired outcome

Sometimes learners are given a problem to solve and then left to work in an open-ended manner until they find out for themselves.

This is one of the most suitable methods of delivering a competence-based curriculum.

(c) Group or class discussion or pair work

In this method, the teacher and learners interact through question and answer sessions most of the time. The teacher carefully selects his questions so that learners are prompted to think and express their ideas freely, but along a desired line of thought. Discussion method should take learners from known to unknown in a logical sequence; and works well with small groups of learners. The disadvantage of this method is that some learners maybe shy or afraid to air their opinions freely in front of the teacher or their peers. This makes them passive participants while the more confident learners dominate the discussions. However, the method enhances learners interpersonal and communication skills.

d) Project method

In this method, the teacher organizes and guides a group of learners or the whole class to undertake a comprehensive study of something in real life over a period of time such as a week or several weeks.

Section 2: Pedagogical approaches

Learners using the project method of studying encounter real-life problems which cannot be realistically brought into a normal classroom situation. A project captures learners' enthusiasm, stimulates their initiative and encourages independent enquiry. The teacher, using the project method, must ensure that the learners understand the problem to be solved and then provides them with the necessary materials and guidance to enable them carry out the study. The teacher can use the project method for topics, which cannot be adequately studied during the normal time-tabled school periods.

Disadvantages

If a project is not closely supervised, learners easily get distracted and therefore lose track of the main objective of their study. Studying by the project method does not work well with learners who have little or no initiative.

(e) Educational visits /excursions

This is a lesson conducted outside the school compound during which a teacher and the learners visit a place relevant to their topic of study. A suitable educational visit for and Computer Science learning is a visit to an software development firm. Such visits enable learners to knowledge and skills that cannot be acquired in a classroom setting. It also

allows them to learn practically through first-hand experience. In such visits, learners are likely to be highly motivated and the teacher should exploit this in ensuring effective learning. However, educational visits are time-consuming and require a lot of prior preparation for them to succeed. They can also be expensive to undertake especially when learners have to travel far from the school.

(f) Demonstration lessons

In a demonstration, the teacher shows the learners the procedure to be followed when performing a particular task. The learners gather around the teacher where each learner can observe what the teacher is doing. It is necessary to involve the learners in a demonstration by:

- Asking a few learners to assist you in setting up the apparatus.
- Asking them to make observations and ask questions where it
- Asking them questions as you progress with the demonstration.

This will help to prevent the demonstration from becoming too teacher-centred.

When is a demonstration necessary?

A teacher may have to use a demonstration, for example when:

- The experiment/procedure is too advanced for learners to perform.
- The task/ procedure is dangerous
- The apparatus and materials involved

are delicate for learners to handle.

- Apparatus and equipment are inadequate.

Importance of practical work in the learning Computer Science concepts

Most of Computer Science concepts are learned through practical hands on activities and interaction with IT tools and devices. For example learning how to work with application soft wares and programming. Therefore, the learners should be involved in practical activities as much as possible. Remember the old Chinese proverb that says:

- What I hear I forget
- When I see I remember
- When I do I understand

2.2 Teacher's role in teaching and learning of Computer Science

The change to a competence-based curriculum is about transforming learning, and ensuring that learning is deep, enjoyable and habit-forming. Teachers must shift from the traditional method of instruction and play the role of a facilitator. They must let learners experience the content through the use of available resources in order to value each learner's individual needs and expectations.

The teacher must identify the needs of learners, the nature of the learning to be done, and the means to shape learning experiences accordingly.

A teacher's role is to organise the learners in the classroom or computer laboratory and to engage them in the use of these learning through participatory and interactive methods This ensures that the learning is personalised, active, participative and cooperative.

The teacher will design and introduce the tasks to the class to perform or for immediate discussion. The role of the teacher will be to guide the learners in constructing their own knowledge, skills and attitudes. Learners are taught how to use the computer and other IT tools used in different technologies.

The teacher must select and develop appropriate materials like teaching models, digital content, and simulators for the learners to use in their work. In practical lessons, the teacher first demonstrates the handling of the computer and other IT tools followed by learners embarking on the task with hands on manipulation.

The teacher must devise remedial strategies in and outside the classroom/ computer laboratory to ensure low achievers and those with learning difficulties keep pace with the rest in acquiring the required competencies.

The following are some ways through which the teacher may facilitate learning during lesson execution:

- Encouraging and accepting student autonomy and initiative;
- Allowing student responses to drive lessons, shift instructional strategies,

and alter content;

- Encouraging students to engage in dialogue, both with the teacher and one another;
- Motivating learners to make them ready for learning.
- Coordinating learners' activities so that the desired objectives can be achieved.
- Appreciating, evaluating and correcting the students' responses before sharing their own understandings of those concepts;
- Engaging students in experiences that pose contradictions to their initial hypotheses and then encouraging discussion;
- Nurturing students' natural curiosity.
- Assessing learners' activities and suggest solutions to their problems.
- Assisting learners to consolidate their activities by summarising the key points learnt.

2.3 Learner's role in learning Computer Science

The activities of the learner are suggested in each learning unit and reflect the appropriate level of engagement of the learner in the learning process. The teaching and learning processes should be tailored towards creating a learner friendly environment based on their capabilities, needs, experience and interests.

The learning activities should be organised in a way that encourages

learners to construct knowledge either individually or in groups in an active way. Learners should work on one competence at a time in the form of concrete units with specific learning outcomes broken down into knowledge, skills and attitudes.

For active participation in competence based learning, of Computer Science concepts, the learners should given a chance to:

- Interact with IT equipment in order to acquire hands on experience
- Raise questions about what is observed, suggest solutions to those questions and carry out investigations to verify the appropriateness of the solutions.
- Apply the acquired knowledge and skills to solve real life problems.
- Working corroboratively with others, communicating their own ideas and considering others' ideas; Take part in planning investigations with appropriate controls to answer specific questions;
- Engaging in lively public discussions in defense of their work and explanations;

2.4 Grouping learners for learning

Most Computer Science activities are carried out in groups During such activities, the teacher should guide the learners to in arranging their desk in order to sit together in groups

In certain activities, the teacher may wish to carry out a demonstration.

In this case, the learners should be sitting or standing in a semi-circle, or arranged around an empty shape of letter “U” such that each learner can see what the teacher is doing clearly and without obstruction or pushing.

The following are some criteria used when grouping learners:

- (a) Similar ability grouping.
- (b) Mixed ability grouping.
- (c) Similar interests grouping.
- (d) Grouping based on gender.

The most suitable criteria is the Mixed ability grouping that.

Grouping learners has several advantages such as:

- (a) Learners can learn from one another.
- (b) Materials that were inadequate for individual work can now easily be shared.
- (c) A teacher can easily attend to the needs and problems of a small group.
- (d) Cooperation among learners can easily be developed.
- (e) Many learners accept correction from the teacher more readily and without feeling humiliated when they are in a small group rather than the whole class.
- (f) Learners’ creativity, responsibility and leadership skills can easily be developed.

There is no fixed number of learners that a group must have. This is dictated by such factors as the task to be done, the materials, characteristics of learners in your class, size and the space available.

However, groups should on average have between four to seven learners. You can also resort to **pair work** depending on the nature of the content being taught at the time.

2.4 Teaching Resources

The successful implementation of this computer science syllabus require learners to have computers which are standard equipment for this subject. Alongside computers, other computing tools and application software have been identified and it is indicated in the syllabus where they will be needed.

Various resources for the implementation of competence-based computer science are the following:

- **Computer laboratory:** The allocation should be one computer for one student. It is recommended to school to explore available technologies in the market to minimise cost of lab equipment.
- **Peripheral devices:** printer, scanner, digital camera, headphone, microphone and speakers.
- **Telephony systems:** Smartphone during Android operating systems lessons
- **Projectors:** Presentation is the key element of competence based

curriculum where students work. Teaching materials will be mostly displayed using a projector for computer science content.

- **Laptop per teacher:** Teachers needs to prepare learning and teaching materials and organise content so as to use the classroom time effectively. One laptop per teacher is required.
- Internet connectivity.
- **Internet connectivity:** Wired or wireless internet connection in School for research
- **Network devices:** Switch, router, wireless access point, server, Cat6 cables, networking toolkit and cabling duct.
- **Software:** In most cases skills expected from this competence based curriculum do not rely on any version of operating system or any version of application software. However, the latest version of most software at the time of implementation will be used.
 - Operating system: Licensed copy of the Windows Operating System. Linux may be used if all applications reflected in the syllabus are supported.
 - ERD commander and paragon recovery software.
 - Microsoft Office (Word, Excel and PowerPoint for

research, producing notes and presentation. , spreadsheets and presentation software.

- Visual Studio for Visual Basic and C++ programming. You may also use Turbo C++ or open source compiler
- Database platform: Licensed copy of Access and MySQL database management systems (DBMS). Consider using XAMPP for MySQL.
- Netbeans or Eclipse IDE
- Browsers: Chrome, Mozilla Firefox, Epiphany and Opera.
- Boson Network simulator, Cisco packet tracer simulator.
- Adobe Photoshop image editor, GoldWave Sound Editor Windows Movie Maker/Corel Video Studio, Macromedia Flash, and Adobe Creative Suite.

2.5 Human resource

The effective implementation of this curriculum requires a joint collaboration of educators at all levels. Given the material requirements, teachers are expected to accomplish their noble role as stated above. School head teachers and directors of studies are required to follow-up and assess the teaching and learning of computer science. These combined efforts will ensure

bright future careers and lives for learners as well as the contemporary development of the country.

In a special way, computer science teacher at advanced level should have a firm understanding of computer science concepts at the level he/she teaches. He/she should be qualified in Information technology/ Computer Science or related fields with Education and have firm ethical conduct. The teacher should possess qualities of a good facilitator, organiser, problem solver, listener and adviser. He/she is required to have skills and competency of guidance and counseling.

2.6 Skills required for computer science teacher

The teacher of computer should have the following skills, values and qualities:

- Inspire children and community the devotion of learning and using computers.
- Engage learners in variety of learning activities.
- Use multiple teaching and assessment methods.
- Adjust instruction to the level of the learners.
- Use creativity and innovation in the teaching. Be a good communicator and organiser.
- Be a facilitator and a counselor.
- Manifest passion and impartial love for children in the teaching

and learning process.

- Link computer science with other subjects and real life situations.
- Have good mastery of computer science content and devotion to be a lifelong learner.
- Have good classroom management skills.

2.7 Classroom as a learning environment

A classroom refers to the place where learning takes place. Students learn from everything that happens around them, such as the things that they hear, see, touch, taste, smell and play with. It is therefore important for the teacher to make his classroom an attractive and stimulating environment. This can be done by:

- Carefully arranging the furniture and desks
- Putting up learning and teaching aids on the walls. Examples are wall charts or pictures or photographs.

• Securing the storage area

Classroom organisation

A well-organised classroom is an asset to good teaching and learning.

The teacher should consider the following factors when organising the classroom for Computer Science lesson:

- (a) Furniture should be well arranged so as to allow free movement of learners and the teacher.

- (b) The number of learners should be properly matched to the available resources.
- (c) Learners should be reasonably spread out so that they do not interfere with one another's activities.
- (d) The series of lessons or activities going on for a number of days or weeks such as individual or group work or whole class.
- (e) There should be adequate light in the classroom to enable all learners including those with visual

challenges to see the chalkboard and instructional materials

- (f) In a computer room, all safety procedures should be displayed on a wall chart, and safety be considered in the design and organisation of the room.

Some tasks where safety measures should be strictly observed include:

- Cabling ICT devices and connecting them to the power supply
- When performing diagnostic tasks inside the system unit,

Section 3: Planning for teaching

The two most important documents in planning to teach are the schemes of work and the lesson plan.

3.1 Schemes of work

A scheme of work is a collection of related topics and subtopics drawn from the syllabus and organized into periods week by week for every term. It is also a forecast or plan that shows details under these subheadings:

- Week
- Key unit competency
- Period
- Learning objectives
- Learning resources and reference materials
- Teaching methods and techniques
- Observations/self evaluation
- Comments from school director (DOS)

In addition, the schemes of work shows the day when a specific lesson will be taught and how long it is intended to take.

Week - refers to the week in the term e.g. 1, 2, 3, etc.

Key unit competency - Gives the competence learners are expected to achieve at the end of the unit.

Lesson - refers to the lesson being taught in that week e.g. lesson 1,2,3 and 4, etc. This shows which is a single and which is a double lesson.

Date - the day when the lesson will be taught.

Sub-topic - a subset of the topic which is a smaller component of the unit e.g. under the topic plants, one could have 'parts of a plant' as a sub-topic.

Objective - what pupils are expected to achieve at the end of the lesson.

Learning resources - any materials that will be used by the pupil and the teacher for learning and teaching.

References - books or other materials that will be consulted or used in the teaching process. Books that pupils will use should also be shown here; indicating the actual pages.

Observations/self evaluation - this should be a brief report on the progress of the lesson planned in the scheme of work. Such reports could include: 'taught as planned'. Not taught due to abrupt visit by Country Director of Education.' 'Children did not follow the lesson, it will be repeated on... (Specific date).

Comments from director of school – space left for comments by the school director.

Sample Scheme of work

Academic year: 2016

Term: I

School: GS NDERONZIZA

Subject: Computer Science

Teacher's name: KAMANA Clement

Class: Senior 4

Week	Key Unit Competencies	Lessons	Learning objectives	Resources & References	Teaching methods & techniques	Observations/ self-evaluation (include dates of assessment)
Week 1 (2-5 Feb)	Explain characteristics and evolution of computers and detect the impact of computer in society	<p>Lesson 1: Definition of a computer and computer science. Computer characteristics.</p> <p>Lesson 2: Classification of computers according to size and processing power</p> <p>Lesson 3: Classification of computers according to data type</p>	<p>Learners should be able to define a computer and computer science.</p> <p>Learners should be able to state characteristics of computers.</p> <p>Learners should be able to classify computers according to physical size and processing power</p> <p>Learners should be able to classify computers according to data type</p>	Student's book, computer, projector, white board, printer, internet, video clips and pictures showing type of computers.	Group work activities, research, question and answer, practical activities, field excursions, demonstration, and observation.	<p>I did not cover computer characteristics on 2nd Feb being the first day of the term.</p> <p>Strategy: Lesson to be covered during free time on 9th Feb.</p> <p>Classification of computers size and processing power covered satisfactorily.</p>
Comments from Director of School						

3.2 Lesson plan

A lesson plan is a detailed outline of how the teacher intends to carry out a specific lesson.

Important sub-headings of a Lesson Plan

1. Administrative details

Date..... Subject.....

Class.....

Time..... Roll.....

2. Topic area

Broad area that is to be studied, taken from the syllabus.

3. Sub-topic area

A smaller topic of the topic about which a lesson will be taught.

4. Key unit competence

This is/are the competence(s) that the learner is expected to achieve at the end of the unit.

5. Learning Objectives

These represent what the teacher anticipates pupils to achieve by the end of the lesson. Objectives should be clear and specific. They should also be stated in behavioural terms, that is, in a way that the outcome can be seen, displayed or measured. In one should distinguish between knowledge, skill and attitude objectives.

6. Learning/teaching resources

Any materials and apparatus that the pupils and the teacher will use during the lesson.

7. References

Any resources consulted or used by the teacher to prepare the lesson

as well as any books that the pupils will use during the lesson.

8. Introduction

This is the start of the lesson. The teacher should motivate the pupils by creating learning situations that interest pupils e.g. posing a problem, telling an amusing but relevant story or episode, showing an object or picture that arouse their interest. The introduction should link what the pupils have already learnt with what they are going to learn.

9. Presentation/lesson development

This should mainly include the activities that pupils and the teacher will perform in order to achieve the stated objectives; as well as the questions that pupils will answer as they do the various activities.

It is convenient to distinguish between the pupils' and teacher's activities under two columns.

10. Summary/conclusion: (Consolidation)

This is the step in which the lesson activities are tied up or consolidated to emphasize the main points, summarize the lessons or make conclusions. The summary should correspond to the objectives stated for that lesson.

11. Comments/self-evaluation:

Teacher should write remarks on whether the objectives were achieved or not and what he or she intends to do to improve on the weak points noted during the lesson.

Sample Lesson Plan

School Name: GS NDERONZIZA

Teacher's name: KAMANA Clement

Term	Date	Subject	Class	Unit No	Lesson No	Duration	Class size
I	4/02/ 2016	Computer Science	S4	I	I of 6	40 minutes	35
Type of special needs to be catered for in this lesson and the number of learners in each category				Special Cases in class: Visually challenged No: I: not blind but required to sit in front Hearing challenged No: none Physically challenged No: I to be assisted where mobility is required			
Topic area:	Computer System and Maintenance						
Unit title	Computer Fundamentals						
Key Unit Competence:	Explain characteristics and evolution of computers and detect the impact of computer in society						
Title of lesson	Classification of Computers According to Size and Processing Power						
Instructional Objectives	Through group discussions the learner, should be able to classify computers according to physical size and processing power						
Classroom setup	Inside the classroom or laboratory.						
Learning Materials	Pupils' textbooks, charts on various computer classes, pictures of super, mainframe, mini and micro-computers.						
References	Pupils' textbooks, Teacher's guide, any other reference book or websites on classification of computers according to physical size and processing power.						

Timing for each step	Description of teaching and learning activity		Competences and cross-cutting issues to be addressed
	Through activity in groups or pairs, research from the internet on types of computers according to size and processing power. The learners should also be able to identify characteristics of supercomputers, mainframe, minicomputers and micro computers. This will help learners appreciate importance of each type of computer and its use.		
	Teacher activities	Learner activities	

<p>Introduction (5min)</p>	<p>Guiding learners to:</p> <ul style="list-style-type: none"> Form groups and carry out activity 1.4 on page 3 in student's book. Guiding learners to perform activities 1.5 to 1.8 from page 4-6 	<p>Groups to do research on types of computers as required by activity 1.4. Each student to participate in group discussions</p>	<p><i>Cooperative learning</i> will be enhanced achieved as the learners work in groups</p>
<p>Development (30min)</p>	<p>As the facilitator: during the lesson:</p> <ul style="list-style-type: none"> Teacher takes control and request specific group leaders to present their findings to entire class Moderates the discussion that comes up Explains various types of computers according to size and processing power Uses the pictures and video clip show type of computers not available in the school 	<p>Learner to carry out activities 1.5 to 1.8 pairwise or in groups.</p> <p>The group leader to makes presentation to the class</p> <p>Learners to identify each type of computer from pictures and video clips provided by the teacher.</p>	<p><i>Inclusivity in education</i> - Learner who is visually challenged is given a chance to sit in front of the class.</p> <p><i>Critical thinking</i> This will be practiced as learners distinguish types of computers according to size, their characteristics and uses.</p> <p><i>Cooperative learning</i> will be enhanced achieved as the learners work in groups</p>
<p>Conclusion (3min)</p>	<p>Teacher summarizes the lesson by:</p> <ul style="list-style-type: none"> Classifying computers according to size and processing Giving assignment exercise on characteristics and uses of supercomputers, mainframes, minicomputers or PCs.. 	<p>Learners states four type of computers according to size.</p> <p>Learners carry out research on uses of of supercomputers mainframes, minicomputers and PCs.</p>	<p>Listening skills will be enhanced as learners take final notes.</p> <p><i>Communication skills</i> will be enhanced as learners answer questions in the assignment.</p>
<p>Evaluation (2min)</p>	<p>Evaluating the success of the lesson by asking learners to quickly identify types of computers according to physical size.</p>	<p>Asking and responding to questions</p>	<p><i>Communication skills</i> will be enhanced as learners answer oral questions.</p>

Section 4: Assessment and record keeping

Assessment is the process of evaluating the teaching and learning processes through collecting and interpreting evidence of individual learner's progress in learning and to make a judgment about a learner's achievements measured against defined standards. Assessment is an integral part of the teaching and learning processes. In the new competence-based curriculum assessment must also be competence-based; whereby a learner is given a task and asked to do it by applying what he/she learned. The two types of assessment that are employed in the new curriculum are **formative** and **summative** assessment.

4.1 Formative and continuous assessment

Formative or continuous assessment involves formal and informal methods used by schools to check whether learning is taking place. When a teacher is planning his/her lesson, he/she should establish the criteria for performance and behavior changes at the beginning of a unit. Then at the end of every unit, the teacher should ensure that all the learners have mastered the stated key unit competencies basing on the criteria stated, before going to the next unit. The teacher will assess how well each learner masters both the subject matter and the generic competencies described in the syllabus and from this. The teacher will get a picture of the all-round progress of

the learner. The teacher may use one or a combination of the following teaching methods:

- Question and answer
- Practical activities
- Observation
- Projects

(a) Question and answer

This method involves asking learners questions during the teaching/learning sessions where the learners are required to respond orally or in writing. Depending on the topic, the questions may be short answer type questions, structured type questions, filling blanks, multiple choice questions, true-false questions and matching items.

(b) Practical activities

This teaching method requires the learner to perform a task or solve a problem practically. The teacher then assesses the finished work by looking at the materials used, procedures followed, whether it works or not or whether it is finished. He or she then awards marks accordingly.

(c) Observation

This method involves the teacher observing learners as they perform a task such as programming to assess acquisition of skills and attitude change. The teacher checks the learner's ability to measure, classify, communicate findings, etc. He or she also assesses the learner's curiosity, patience, team

and co-operation spirit among others.

(d) Projects

This involves giving individual learners or group of learners a comprehensive task that they do undertake a comprehensive in real life that they will do over a period of time. The time allowed for the project may span over days, weeks or months after which they present a report. In carrying out a project, let learners begin from planning stage (come up with a schedule of events), execute the plan, analyse the results and look back (reflect on the challenges encountered during the project and come up with solutions to those challenges (problem-solving skills)).

A teacher can use one or several of these assessment methods depending on the subtopic being studied or the purpose for which assessment is required.

4.2. Summative assessment

When assessment is used to record judgment of the competence or the performance of the learner, it serves a summative purpose. Summative assessment provides a picture of a learner's competence or progress at any specific moment. The main purpose of summative assessment is to evaluate whether learning objectives have been achieved. It is also used for the ranking or grading of learners, for deciding on progression, for selection into the next level of education, and for certification.

This assessment should have an

integrative aspect whereby a student must be able to show mastery of all competencies.

It can be internal/school-based or external in the form of district and national examinations. School-based summative assessment should take place once at the end of each term and once at the end of the year.

Before developing a question paper, a plan or specification of what is to be tested or examined must be produced to show the units or topics to be tested on, the number of questions in each level of Bloom's taxonomy, and the marks allocation for each question.

In a competency based curriculum, questions from the higher levels of Bloom's taxonomy should be given more weight than those from the knowledge and comprehension level. Before developing a question paper, the item writer must ensure that the test or examination questions are tailored towards competency based assessment by doing the following:

Identify topic areas to be tested on from the subject syllabus.

Outline subject matter content to be considered as the basis for the test.

Identify learning outcomes to be measured by the test.

Prepare a table of specifications.

Ensure that the verbs used in the formulation of questions do not only require memorisation or recall answers

but test the broad competencies as stated in the syllabus.

4.3 Structure and format of the examination

1.2.2 Structure and format of examination

In computer science, two types of papers are to be examined: theory paper and practical paper.

- Theory paper tests knowledge, skills and attitudes. The areas to be examined for competence include Computer systems and maintenance, data structures and algorithms, programming, databases, networking, computer graphics and multimedia.
- For programming, database and networking, students are supposed to develop a project using specified criteria
- The practical paper that consists of question on transforming an algorithm into a computer program using C/C++, Visual Basic and java; database questions and networking.
- The content distribution table provided in the syllabus will assist in making choice of the content to be assessed because not all topics and subtopics can be assessed.

4.4 Record Keeping

Record keeping is the routine of managing student's performance and teaching progress. One of the most

important documents in keeping records of assessments and teaching is the portfolio,

A Portfolio is a folder (or binder or even a digital collection) containing the student's work as well as the evaluation of the student's strengths and weaknesses in tasks given.

A Portfolio reflect not only the work produced, but is also a record of the activities undertaken over time as part of teaching and learning process. The portfolio output (formative assessment) will be considered only as enough for three years of Advanced level. Besides, it will serve as a verification tool for each learner that he/she attended the whole learning before he/she undergoes the summative assessment for the subject.

Whatever procedures are used in formative assessment, generate data in form of competences and record them in each student's portfolio. This is useful in monitoring learning progress of each learner for purposes of remedial actions, alternative instructional strategy, feedback/advice to learner and parents.

4.5. Reporting to parents

The wider range of learning in the new curriculum means that it is necessary to think again about how to share a learners' progress with their parents.

A single mark is not sufficient to convey different expectations of learning that are in the learning objectives. The most helpful reporting is to share what learners are doing well, and where they need to improve. A simple scale of meeting/not meeting

expectations very well, for each of knowledge/understanding, skills and competencies in a subject will convey more meaning than a single mark. For school based assessments, these scores do not need to be added up.

	UNIT 1 Computer Fundamentals	UNIT 2 Computer Architecture and Assembly	UNIT 3 Safe and ethical use of computers	UNIT 4 Computer Software Installation	UNIT 5 Number Systems	UNIT 6 Boolean Algebra and Logic Gates
Number of periods	16	18	12	16	18	18
Introduction	<p>This unit covers the following key topics:</p> <ul style="list-style-type: none"> • Definition of computer science • Characteristics of computer • Classification of computers • Role of computers in the society • Historical evolution of computers 	<p>This topic covers the following key topics:</p> <ul style="list-style-type: none"> • Computer system. • Peripheral devices. • System unit (case) types. • Ports and connectors. • Assembling computers. • Cleaning and disposing of computer components. 	<p>This topic covers the following key topics:</p> <ul style="list-style-type: none"> • Ergonomics and structures furniture. • Safely against climate change. • Protection against physical damage. • Protecting the environment. • Computer ethics. 	<p>This topic covers the following key topics:</p> <ul style="list-style-type: none"> • Computer software • Disk preparation • Booting order • Installing drivers • System requirements • Disk management • Installing OS • Installing application software. 	<p>This topic covers the following key topics:</p> <ul style="list-style-type: none"> • Fundamentals of number systems • Number base systems • Converting decimal to base-n system • Converting binary to base-n system • Conversion from octal to decimal • Conversion from hexadecimal to decimal • Conversion from decimal fraction to binary • Conversion from binary fraction to decimal • Conversion from negative decimal to binary • Binary arithmetic 	<p>This topic covers the following key topics:</p> <ul style="list-style-type: none"> • Introduction. • Logic gates and circuits. • Introduction to website design. • Boolean algebra. • Animation using Adobe Flash.

Classroom Organisation	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work – activities • Practicals
Equipment Required	<ul style="list-style-type: none"> • Student text book • Projector • Computer • Internet • connectivity • Library • CD/DVD • Video Player • Dismantled computer 	<ul style="list-style-type: none"> • Student text book • Projector • Computer • Internet • connectivity • Library • CD/DVD • Video Player • Dismantled computer 	<ul style="list-style-type: none"> • A computer running Windows OS • Student text book • Projector • Internet • connectivity 	<ul style="list-style-type: none"> • Computer • Fire extinguisher • Dismantled computer • components • Cleaning brushes, soft towels • Projector • Screw drivers • Vacuum cleaners • Blower • Student text book 	<ul style="list-style-type: none"> • Computer • Projector • Scientific calculators 	<ul style="list-style-type: none"> • Student text book • Projector • Computer • running word processor • Internet • connectivity • School and Public library
Activities	<p>Group work:-</p> <ol style="list-style-type: none"> Definition of computer and computer science- 1.1,1.2 Characteristics of computers- 1.3 Classification of computers- 1.4 Uses of computers 1.5 	<p>Group work:-</p> <ol style="list-style-type: none"> Computer system – 2.1 .System unit – 2.3 Definition of ports and connectors – 2.5 – 14 Internal computer components – 2.13 - 14 	<p>Group work:-</p> <ol style="list-style-type: none"> General safety guideline. - 3.1 Environmental issues – 3.2 Computer ethics – 3.9 – 3.6 	<p>Group work:-</p> <ol style="list-style-type: none"> Classification of computers – 4.2, software installation – 4.4 - 4.7 	<p>Group work:-</p> <ol style="list-style-type: none"> Fundamentals of number systems – 5.1 Number base systems – 5.3, 5.4 	<p>Group work on:-</p> <ol style="list-style-type: none"> Logic gate structure - 6.2 Truth tables - 6.5, 6.6 <p>Individual work:--</p> <ol style="list-style-type: none"> Introduction- 5.1 Logic circuit 6.12

	<p>(e) Types of computers according to function, -1.9 data type – 1.10 (f) Role of computers in the society – 1.11 (g) Historical evolution of Computers – 1.12 to 1.16 Assignment - On types of computers. Individual work:-</p>	<p>e) Assembling computers: components – 2.15 – 2.26 Pair work:- a) Peripheral devices – 2.2 b) Ports and connectors – 2.4 c) Cleaning and disposal of computer components – 2.17 - 2.29</p>	<p>d) Fire safety guidelines – 3.3 e) Safety against climate change – 3.5 f) Electrical safety – 3.4 - 3.5</p>	<p>Pair work:- a) Classification of computers – 4.1, b) software licencing – 4.3 c) Disk mangement – 4.5, 4.6</p>	<p>Individual work:- a) Number base systems – 5.2, 5.3, b) Number base conversions -5.4 to 5.12 c) Binary arithmetic – 5.12 to 5.20</p>	<p>c) Truth tables 6.3, 6.12, 6.14 d) Boolean algebra– 6.7 Pair work a) Truth tables 6.9 b) Boolean algebra 6.9</p>
--	---	--	--	--	--	--

Competences Practiced	<ul style="list-style-type: none"> • Critical thinking • Communication skills • Research and problem solving • Cooperation and interpersonal management life skills • Co-operation 	<ul style="list-style-type: none"> • Critical thinking • Communication and research • problem solving • Cooperation and interpersonal management • Life skills 		<ul style="list-style-type: none"> • Hands on skills of cleaning the computer; using fire extinguisher. • Problem solving: troubleshooting 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Science and technology • Creativity and Innovation 	<ul style="list-style-type: none"> • Problem solving • Learning through doing • Cooperation and interpersonal management life skills
Language Practice	<ul style="list-style-type: none"> • Mastery of terms used to describe computer 	<ul style="list-style-type: none"> • Mastery of the names of various parts of a computer 	<ul style="list-style-type: none"> • Use of correct terms in describing safety measures ethical and unethical use of computers 	<ul style="list-style-type: none"> • Mastery of names of various categories of software. • Terms describing the tasks performed during software installation 	<p>Master of names of various numeration system</p>	<ul style="list-style-type: none"> • Mastery of logic terms • Proper communication of logic facts
Vocabulary Acquisition	<ul style="list-style-type: none"> • Understanding computer fundamentals terminologies 	<ul style="list-style-type: none"> • Knowing the names of various devices 	<ul style="list-style-type: none"> • Learning names of various features on the desktop 	<ul style="list-style-type: none"> • Names of various tools and resources 	<ul style="list-style-type: none"> • Understanding number systems concepts 	<ul style="list-style-type: none"> • Mastery of various Boolean algebra terms
Numeracy					<ul style="list-style-type: none"> • Conversion of number systems. • Boolean algebra • Binary arithmetic 	<ul style="list-style-type: none"> • Working out Boolean algebra truth tables. • Solving expressions.
Study Skills	<ul style="list-style-type: none"> • Ability to search for information on the internet. • Ability to summarise main 	<ul style="list-style-type: none"> • Be able to classify a device as input, output, storage or processing. 	<ul style="list-style-type: none"> • Identify each device provided or in pictures and give its name & function. 	<ul style="list-style-type: none"> • Learning by doing: all the practical activities in this unit expose the learner to important tasks. 	<ul style="list-style-type: none"> • Research skills to find out reasons why certain practical implementations 	<ul style="list-style-type: none"> • Computational skills • Research skills

	<ul style="list-style-type: none"> points from a discussion. Writing skills. learning by doing 	<ul style="list-style-type: none"> Explain how each device functions e.g. a 3D printer or an LCD monitor etc. Connect peripheral devices to the computer – match cable to port correctly Identify various ports on the system unit. Classify computers Elucidate the history of computers 	<ul style="list-style-type: none"> Research skills: accessing the internet to find out more about various tasks Analysis skills: demonstrate skill in analysing which storage device to use when. 	<p>are preferred e.g. why prefer a CO2 fire extinguisher to a water/powder based?</p>	<p>skill in analysing analog and digital technologie</p>	<ul style="list-style-type: none"> Research: searching for information on the web and Help. Numeracy: working out truth tables. Creative: designing digital circuits.
Revision	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises 	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises 	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises 	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises 	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises 	<ul style="list-style-type: none"> Provision of both practical and theoretical exercises
Assessments	<ul style="list-style-type: none"> A formative assessment of competencies in each of the core areas covered by this unit. 	<ul style="list-style-type: none"> Assess ability of the learner to classify various devices as input, storage, output or processing 	<ul style="list-style-type: none"> Formative assessment of the following:- ability to identify input/output/storage/ 	<ul style="list-style-type: none"> Assess the ability of the learner to identify various desktop features and their functions by name. 	<ul style="list-style-type: none"> A formative assessment of competencies in each of the core areas covered by this unit. 	<ul style="list-style-type: none"> A formative assessment of competencies in performing the tasks in the student book.

	<ul style="list-style-type: none"> Assess ability to classify computers Assess knowledge of computer evolution according to generation 	<ul style="list-style-type: none"> processing devices and explain how they function. Formative assessment of students ability to identify ports and connect devices. Cleaning the computer Assembling the computer 	<ul style="list-style-type: none"> Formative evaluation of the practical skills of the student to: <ul style="list-style-type: none"> delete/restore files customise start menu set time/date create shortcuts start and close programs Use accessories Assess ability to identify and use various removable storage devices for particular contexts Ability to customise features using control panel 	<ul style="list-style-type: none"> The learner should be able to individually do the practical tasks in this unit - - - even in situations where the there is group work – the teacher must concentrate on the individual. 	<ul style="list-style-type: none"> Assess ability to perform different number systems conversions Assess ability to perform binary arithmetic 	<ul style="list-style-type: none"> Assess ability work out truth tables and simplify Boolean expressions.
Learning Outcomes	<ul style="list-style-type: none"> Concise definition of a computer Concise definition of computer science List characteristics of a computer List characteristics of various classes of computers Demonstrate understanding in classifying 	<ul style="list-style-type: none"> Student classify devices as input/ processing/output/ storage without fail Student to demonstrate skill to attach peripheral devices to the system unit using the right cable and port Student to know devices meant to cater for challenged users 	<ul style="list-style-type: none"> Student should be able to start/ shut down the computer using correct procedures Student to observe safety guidelines strictly Student to demonstrate understanding of computer room ergonomics 	<ul style="list-style-type: none"> Student to be able to define computer software. Student to be able to differentiate between system and application software. Set the booting order in order to install the OS from disk. Perform administrative functions like scandisk 	<ul style="list-style-type: none"> Concise definition of a bit, nibble, byte and word Conversion of number from one system to another Use ones and twos complements Perform binary addition, subtraction, multiplication and division 	<ul style="list-style-type: none"> Student must be able to: <ul style="list-style-type: none"> identify logic operators, Truth tables and evaluate Boolean expression using Boolean laws. Describe the use of logic gates and apply Boolean laws on logic gates.

	<p>computers according to physical size and power.</p> <ul style="list-style-type: none"> • Demonstrate understanding in classifying computers according to function. • Demonstrate understanding in classifying computers according to data type. • State roles of computers in the society • Elucidate historical development of computers. 	<ul style="list-style-type: none"> • Student to assemble the computer from scratch. • Student to identify internal computer components • Student to clean the computer correctly using the right detergents and equipment. 	<ul style="list-style-type: none"> • student to demonstrate knowledge of good furniture • Student to know how to take care of special needs people. • Student to know how to operate a fire extinguisher • Student to observe fire safety procedures • Student to protect computers and components from electrostatic damage • Protect computers from power fluctuations • Protect the environment from e-Waste contamination • Student to adhere to ethics: genuine software; avoid counterfeits. 	<ul style="list-style-type: none"> • Student to be able to install the OS on a blank computer – after partitioning and formatting the hard disk. • Install application software • Install drivers 	<ul style="list-style-type: none"> • Apply laws of Boolean algebra on Boolean expressions. • Draw graphical representation of different logic gates and construct and evaluate a logic circuit.
--	---	---	--	--	---

	UNIT 7 Introduction to Computer Algorithm	UNIT 8 Control Structures and Arrays	UNIT 9 Introduction to Computer Programming	UNIT 10 Introduction to C++ Programming	UNIT 11 Operators and Expressions in C++	UNIT 12 Control Statements in C++
Number of periods	24	12	6	12	12	16
Introduction	This unit covers the following key topics: <ul style="list-style-type: none"> • Fundamentals of algorithms • Program development process • Formulating algorithms • Memory concepts: variables and constants • Operators and Expressions • Modular Programming 	This unit covers the following key topics: <ul style="list-style-type: none"> • Control structures • Sequence control structure • Selection control structures • Iteration control structures • Break and continue statements • Goto statements • Exit statement • One dimensional array 	This unit covers the following topics:- <ul style="list-style-type: none"> • Computer programming concepts • History of programming languages • Computer programming paradigms 	This unit covers the following key topics: <ul style="list-style-type: none"> • Evolution of C++ • Syntax of C++ program • Compiling and executing C++ program • Input and Output streams • Variables and data type • Constants • Output formatting 	This unit covers the following key topics: <ul style="list-style-type: none"> • Expressions and operators • Classification of C++ operators • Overall operator precedence • Classification of C++ expressions 	This unit covers the following key topics: <ul style="list-style-type: none"> • Review of control structures • Selection statements in C++ • Looping statements in C++ • Jump statements
Classroom Organisation	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities • Programming exercises 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities • Programming exercises 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities • Programming exercises 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities • Programming exercises
Equipment Required	<ul style="list-style-type: none"> • Projector • Computer • Network connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • Network connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • Network connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • C++ compiler • Internet connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • C++ compiler • Internet connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • C++ compiler • Internet connection • Digital material

<p>Activities</p>	<p>Group work:- a) Fundamentals of Algorithms- 7.1 b) Formulating algorithms-7.10 c) Operators and expressions-7.9 Individual work (a) Fundamentals of Algorithms- 7.2 - 7.6 (b) Operators and expressions-7.7 - 7.8</p>	<p>Group work:- (a) Sequence n control structures - 8.1 (b) Selection control structures - 8.4 (c) Finite and infinite loop - 8.8 (c) Break and continue - 8.9 Individual work:- a) Selection control structures - 8.2, to 8.3 b) Iteration control structures- 8.5 - 8.6 c) One dimensional array-8.10 to 8.11,</p>	<p>Group work:- a) Computer programming concepts -9.1 and 9.2, b) History of programming languages-9.3 a) Computer programming paradigms-9.6 and 9.7 Individual work:- a) Computer programming paradigms-9.4 and 9.7</p>	<p>Group work:- (a) Syntax of C++ program-10.1 (b) input/output streams - 10.2 (c) Variables and data type-10.6 Individual work:- a) Variables and data type-10.4 and 10.5 (b) Constants- 10.8 (c) Output formatting- 10.11</p>	<p>Group work:- (a) Classification of C++ operators-11.1 and 11.2, (b) Relational operators - 11.5 Individual work:- (a) Types of C++ operators-11.3, 11.4, 11.6 and 11.7, (c) Classification of C++ expressions - 11.8</p>	<p>Group work:- a) Review of control structures -12.1 b) Looping statements in C++ 12.7 Individual work:- a) Selection statements in C++ -12.2 - 12.6, b) Exit statements - 12.8</p>
<p>Competences Practiced</p>	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology
<p>Language Practice</p>	<ul style="list-style-type: none"> • Syntax and semantics of natural languages • Syntax and semantics of algorithms 	<ul style="list-style-type: none"> • Syntax and semantics of control structures • Syntax and semantics arrays 	<ul style="list-style-type: none"> • Syntax and semantics of programming languages • Natural languages 	<ul style="list-style-type: none"> • Syntax and semantics of C and C++ programming language 	<ul style="list-style-type: none"> • Syntax and semantics of C++ operators and expressions 	<ul style="list-style-type: none"> • Syntax and semantics of C++ control statements

<p>Vocabulary Acquisition</p>	<ul style="list-style-type: none"> Mastery of concepts relating to computer algorithms 	<ul style="list-style-type: none"> Mastery of concepts relating to control structures and arrays 	<ul style="list-style-type: none"> Mastery of programming concepts Logic reasoning 	<ul style="list-style-type: none"> Mastery of C++ programming concepts 	<ul style="list-style-type: none"> Mastery of C++ operators and expression 	<ul style="list-style-type: none"> Mastery of C++ control statements
<p>Numeracy</p>	<ul style="list-style-type: none"> Computing mathematical expressions. Computing memory size Boolean evaluation 	<ul style="list-style-type: none"> Computing mathematical expressions. Computing memory size Boolean evaluation 	<ul style="list-style-type: none"> Mathematical expressions Mathematical functions Polish notation Logic expressions 	<ul style="list-style-type: none"> Mathematical expressions Mathematical functions Calculation of memory size 	<ul style="list-style-type: none"> Mathematical expressions Boolean expressions 	<ul style="list-style-type: none"> Mathematical expressions Boolean expressions
<p>Study Skills</p>	<ul style="list-style-type: none"> Analytical skills: formulate algorithms Synthesis: provide solutions to computation problems Computational skills: workout mathematical expressions Research skills: Search for content on internet 	<ul style="list-style-type: none"> Analytical skills: algorithms with control structures Synthesis: solutions to algorithmic structures Computational skills: workout mathematical expressions Research skills: Search for content on internet 	<ul style="list-style-type: none"> Analytical skills: analyze mathematical functions and logic expressions Synthesis: solutions to programming problems Computational skills Research skills: Search for programming material on internet 	<ul style="list-style-type: none"> Analytical skills: analyze C++ arithmetic and boolean expressions Synthesis: solutions to C++ programming problems Computational skills: workout mathematical expressions in C++ Research skills: Search for C++ tutorials on internet 	<ul style="list-style-type: none"> Analytical skills: categorize operator and expression Synthesis: solutions to binary and unary expressions Bitwise operations Logic comparisons Research skills: Search for C++ operators on internet 	<ul style="list-style-type: none"> Analytical skills: analyze boolean expressions Synthesis: solutions to control logic Computational skills Translate conditional logic in algorithms into C++ control statements Research skills: Search for C++ control statements on internet
<p>Revision</p>	<ul style="list-style-type: none"> Provision of activities and theoretical revision 	<ul style="list-style-type: none"> Provision of activities and theoretical revision 	<ul style="list-style-type: none"> Provision of activities and theoretical revision 	<ul style="list-style-type: none"> Provision of practical activities and theoretical revision 	<ul style="list-style-type: none"> Provision of practical activities and theoretical revision 	<ul style="list-style-type: none"> Provision of practical activities and theoretical revision

<p>Assessments</p> <ul style="list-style-type: none"> Formative assessment of definition of algorithms Assess ability to use different algorithm design tools Assess ability to interpret algorithms. Assess ability to formulate algorithms 	<ul style="list-style-type: none"> Formative assessment of definition of control structures Assess ability to define arrays Assess ability to use different selection constructs Assess ability to use iteration constructs. Assess ability to represent one dimensional arrays 	<ul style="list-style-type: none"> Formative assessment of definition of programming languages Assess knowledge on evolution of programming languages Assess knowledge of programming paradigms 	<ul style="list-style-type: none"> Formative assessment of definition C++ concepts Assess ability to differentiate C++ Assess ability to write, compile and run C++ programs Assess ability to identify and use C++ iostream objects and functions Assess ability to declare variables and constants using appropriate data type Assess skills in formatting C++ program output 	<ul style="list-style-type: none"> Formative assessment of definition of operators and expressions Assess ability to identify and use different operators Assess ability to rank operators by order of precedence Assess ability to identify and use different C++ expressions 	<ul style="list-style-type: none"> Formative assessment of definition of control statements expressions Assess ability to identify and use selection statements Assess ability to identify and use selection statements Assess ability to use C++ jump statements like break, continue and goto.
<p>Learning Outcomes</p> <ul style="list-style-type: none"> Definition of algorithms Design of algorithm Interpretation of algorithm 	<ul style="list-style-type: none"> Definition of control structures Design algorithms with control structures 	<ul style="list-style-type: none"> Definition of computer programming concepts 	<ul style="list-style-type: none"> Definition of C++ concepts Knowledge of how C++ languages have evolved 	<ul style="list-style-type: none"> Defining operators and expressions Ability to Categorise operators 	<ul style="list-style-type: none"> Definition of control statements Knowledge of how conditional logic is evaluated

	<ul style="list-style-type: none"> • Problem solving using algorithms • Ability to convert an algorithm to a computer program • Memory management • Modular design of computer programs. 	<ul style="list-style-type: none"> • Interpret algorithms with control structures • Problem solving using control structures • Ability to design ID arrays • Memory management with arrays 	<ul style="list-style-type: none"> • Knowledge of how computer languages have evolved • Definition and categorization of computer programming paradigms • Demonstration of each programming paradigm • Embrace good programming practice • Paradigm shift to object oriented programming 	<ul style="list-style-type: none"> • Master the syntax of C++ programs • Creating, compiling and executing C++ programs • Manipulate input and output streams • Manage memory with variables, data types and constants • Format numeric and non-numeric output 	<ul style="list-style-type: none"> • Ability to Categorise expressions • Formulating unary and binary expression • Demonstrating use of operators • Manipulate decimal numbers and bits • Arrange operators by precedence 	<ul style="list-style-type: none"> • Classification of control statements • Ability to use selection constructs • Ability to use looping constructs • Ability to use jump statements
--	--	--	---	---	--	--

	UNIT 13 Functions in C++ Programming	UNIT 14 Arrays in C++ Programming	UNIT 15 Introduction to Operating System	UNIT 16 HTML-Based Web Development	UNIT 17 Cascading Style Sheet
Number of periods	14	12	18	20	8
Introduction	This unit covers the following key topics: <ul style="list-style-type: none"> • Fundamentals of C++ • Functions • Types of functions • Creating user-defined functions • Recursive functions 	This topic covers the following key topics: <ul style="list-style-type: none"> • Review of arrays • Creating one dimensional array • Accessing array elements • Array of characters • Reading and displaying strings • Creating multi-dimensional arrays 	This topic covers the following key topics: <ul style="list-style-type: none"> • Definition of OS • Functions of OS • Characteristics of OS • Components of OS • Common OS • Smartphone OS • History of OS • Types of OS • MS-DOS 	This topic covers the following key topics: <ul style="list-style-type: none"> • Fundamentals of world wide web • Hypertext Markup Language (HTML) • Introduction to XHTML • Creating web pages • Enhancing we pages • Creating forms • Introduction to HTML5 • Migrating from HTML4 to HTML5 	This topic covers the following key topics: <ul style="list-style-type: none"> • Definition of CSS • HTML styling and disadvantages • Comparison between HTML and CSS styling • CSS syntax • Adding CSS to web pages • CSS Styles • Creating CSS pages from scratch
Classroom Organisation	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities 	<ul style="list-style-type: none"> • Whole class sessions • Group work activities • Practicals
Equipment Required	<ul style="list-style-type: none"> • Projector • Computer • C++ compiler • Internet connection • Digital material 	<ul style="list-style-type: none"> • Projector • Computer • C++ compiler • Internet connection • Digital material 	<ul style="list-style-type: none"> • Student text book • Projector • Computer • Internet connectivity • Library • CD/DVD • Video Player 	<ul style="list-style-type: none"> • Computer • Projector • Java Netbeans IDE • HTML and CSS editor • Internet 	<ul style="list-style-type: none"> • Computer running web server and text editors • Browsers • Student text book • whiteboard • projector • internet connection

<p>Activities</p>	<p>Group work on:-</p> <p>a) Library functions-- 13.1</p> <p>(b) User defined functions - 13.2</p> <p>Individual work:-</p> <p>a) Functions declaration in C++ – 13.3 ab)7</p> <p>b) Function return type and argument - 13.4</p> <p>(c) Parameter passing - 13.5</p> <p>(d) Recursive functions - 13.6 and 13.7</p>	<p>Group work on:</p> <p>(a) initialising arrays and strings- 14.2, 14.4 and 14.5</p> <p>Individual work:-</p> <p>a) Declaration of arrays – 14.1</p> <p>b) Reading and writing array elements l array – 14.3</p>	<p>Group work:-</p> <p>a) Definition of OS – 15.1</p> <p>b) Characteristics of O/S – 15.3</p> <p>(c) Smartphone OS – 15.7</p> <p>Android – 15.8</p> <p>Apple OS – 15.9</p> <p>Windows Phone – 15.10</p> <p>Pald OS – 15.11</p> <p>History – 15.12</p> <p>Pair work:-</p> <p>(a) Component of OS- 15.2, and 15.4</p> <p>Individual:-</p> <p>(a) Types of OS – 15.13 and 15.4</p> <p>MS-DOD – 15.5</p> <p>UNIX-15.6</p> <p>Linux – 15.7</p> <p>Mac OS -15.8</p>	<p>Group work:-</p> <p>a) Fundamentals of world wide web – 16.1</p> <p>b) Hypertext markup language – 16.2</p> <p>c) Hyperlinks and tables - 16.11 and 16.12</p> <p>d) Introduction to HTML5 – 16.15 abd 16.17</p> <p>e) Migrating from HTML4 to HTML5 - 16.16</p> <p>Individual work:-</p> <p>a) Hypertext markup language – 16.3, 16.4</p> <p>b) Creating web pages – 16.5, 16.6</p> <p>d) Creating Forms – 16.13 and 16.14</p>	<p>Group work:-</p> <p>(a) CSS float property – 17.11</p> <p>Pair work:-</p> <p>a) CSS syntax – 17.2</p> <p>Individual work:-</p> <p>a) Definition of CSS – 17.1</p> <p>b) Fonts-and margins 17.5, 17.6 and 17.7</p> <p>(c) Hiding elements 17.9</p> <p>(d) Background - 17.10</p> <p>Practical hand on:-</p> <p>(a) Creating CSS page from scratch – 17.2, 17.4</p> <p>(b) CSS coding strategies- 17.3</p>	<p>Competences Practiced</p> <ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Numeracy • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation • Science and technology 	<ul style="list-style-type: none"> • Problem solving • Learning through doing 	<ul style="list-style-type: none"> • Critical thinking • Cooperation • Interpersonal management • Creativity and innovation 	<ul style="list-style-type: none"> • Critical thinking • Creativity and innovation • Cooperation and interpersonal management • Communication research and problem solving • Practical skills CSS programming
--------------------------	--	---	---	---	---	---	---	---	---	--

Language Practice	<ul style="list-style-type: none"> Syntax and semantics of C++ functions 	<ul style="list-style-type: none"> Syntax and semantics of C++ arrays 		<ul style="list-style-type: none"> Syntax and semantics of HTML4, XHTML and HTML5 	<ul style="list-style-type: none"> Software programming and use Discussion in groups
Vocabulary Acquisition	<ul style="list-style-type: none"> Mastery of C++ concepts relating to functions 	<ul style="list-style-type: none"> Mastery of C++ concepts relating to arrays 	<ul style="list-style-type: none"> Understanding of various operating systems terminologies 	<ul style="list-style-type: none"> Mastery of HTML4, XHTML and HTML5 doctype specifications 	<ul style="list-style-type: none"> Understanding of CSS and HTML
Numeracy	<ul style="list-style-type: none"> Mathematical expressions Boolean expressions 	<ul style="list-style-type: none"> Mathematical expressions Boolean expressions Computing memory size 		<ul style="list-style-type: none"> Determining table size Determine image size 	<ul style="list-style-type: none"> Specifying colours in hexadecimal
Study Skills	<ul style="list-style-type: none"> Analytical skills: analyse functions Synthesis: create user defined functions Computational skills Breakdown a large program into functions Demonstrate parameter passing Research skills: Search for C++ control statements on internet 	<ul style="list-style-type: none"> Analytical skills: analyze arrays Synthesis: create and manipulate 1D arrays Create and manipulate strings Use C++ string libraries Create and manipulate 2D arrays 	<ul style="list-style-type: none"> Learn through practise Research: searching for information on the web and Help typing commands at the DOS prompt 	<ul style="list-style-type: none"> Analytical skills: classify according to HTML doctype Illustrate HTML doc structure Synthesis: create and render HTML documents Create and render XHTML docs Identify differences in HTML, XHTML and HTML5 Create HTML lists, tables and forms Insert graphics into HTML documents Work with hyperlinks Create and render HTML5 documents Use new HTML5 input elements and their attributes 	<ul style="list-style-type: none"> Explain CSS related terms appropriately Program CSS pages

Revision	<ul style="list-style-type: none"> Provision of practical activities and theoretical revision 	<ul style="list-style-type: none"> Provision of practical and theoretical revision 	<ul style="list-style-type: none"> Provision of practical and theoretical revision 	<ul style="list-style-type: none"> Provision of practical and theoretical revision 	<ul style="list-style-type: none"> Provision of practical and theoretical revision
Assessments	<ul style="list-style-type: none"> Formative assessment of definition of functions Assess ability to use library functions Assess ability to create user-defined functions Assess ability to create and manipulate recursive functions. 	<ul style="list-style-type: none"> Formative assessment of definition of arrays and manipulate ID array Assess ability to create array of characters Assess ability to create and manipulate multi-dimensional arrays. 	<ul style="list-style-type: none"> Formative assessment of ability to describe OS functions. Assess ability to differentiate between computer and smart phone OS Assess student understanding of main characteristics of OS Student must be able to describe smart phone OS Assess student on knowledge of history of O/S Assess student on demonstrated knowledge of types of OS Assess student on ability to use MS-DOS commands 	<ul style="list-style-type: none"> Formative assessment of ability to create HTML documents. Assess ability to use XHTML 1.0 Assess student understanding of HTML5 syntax 	<ul style="list-style-type: none"> Demo understanding of the advantages of CSS over HTML formatting of websites. Knowledge of CSS selector and syntax for rule creation. Ability to work with inline, internal and external CSS. Formative assessment task about accurate use of CSS Ability to create completely functioning CSS pages.
Learning Outcomes	<ul style="list-style-type: none"> Ability to define functions Ability to use library functions Knowledge of parameter passing Ability to create functions 	<ul style="list-style-type: none"> Ability to define arrays Ability to create and manipulate ID array Ability to determine size of an array Ability to determine dimension of an array 	<ul style="list-style-type: none"> Student must be able to: <ul style="list-style-type: none"> set page to portrait / landscape Track changes, accept or reject changes insert page, column breaks capability to define and use a style 	<ul style="list-style-type: none"> Appreciate evolution of internet, web and HTML Ability to differentiate HTML, XHTML and HTML Ability to create web pages using HTML Ability to determine appropriate HTML doctype 	<ul style="list-style-type: none"> Differentiate html styling and CSS styling. Identify the correct use of a given selector; how to set a selector and how to name different elements to match one CSS selector.

	<ul style="list-style-type: none"> • Ability to determine function return type • Ability to create recursive functions 	<ul style="list-style-type: none"> • Ability to create and manipulate strings • Ability to create multi-dimensional arrays 	<ul style="list-style-type: none"> • ability to work with nested lists • Insert borders and footers. • insert page numbers • work with drop cap 	<ul style="list-style-type: none"> • Ability to create HTML lists, tables and forms array • Ability add hyperlinks and images to a web page • Manage a smooth transition from HTML4 to HTML5 	<p>Differentiate priorities of styles in a web page namely external CSS, internal CSS and inline CSS.</p> <ul style="list-style-type: none"> • Identify basic properties for different selectors. • Create online styles
--	--	--	---	---	--

Unit I

Computer Fundamentals

Student's Book page 1 – 15. (16 Periods)

Key unit competence

By the end of this unit, the learner should be able to explain characteristics and evolution of computers and detect the impact of computers in society.

Learning objectives

Knowledge and understanding

1. State and explain characteristics of a computers and classify them according to their size, processing power, their functions and data to be processed.
2. Identify the impact of computer in society.
3. Explain the evolution of computer.

Skills

1. Detect the impact of computer in society.
2. Classify computers according to their size, processing power ,their functions and data to be processed
3. Differentiate different computer generations, technology used in each generation.

Attitudes and values

Appreciate the evolution and the importance of a computer in:-

Education
Business
Governance
Health
Communication
Entertainment

Generic competences addressed in this unit

Problem solving: By doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: Working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence supports interpersonal respect and

promotes national unity.

Communication: presentations in class made after group work will develop this skill very well.

Links to other subjects

Physics or electronics related to technologies used in electronic computers: vacuum tubes, transistors, integrated circuits.

Cross cutting issues addressed in this unit

Inclusive Education: This is a teaching methodology that ensures all students are supported to learn regardless of their backgrounds and abilities. It is the responsibility of the teacher to use inclusive approach by integrating parents, community, and educators in his/her teaching methods. As a computer science teacher, make use of available teaching accessibility tools and to help students with disabilities participate in classroom and other aspects of school life.

Assessment criteria

Students can explain the role and evolution of computers, type, classification of computer.

Suggested teaching methodology

Guided discovery

Research

Question and answer

Discussion

Background information

Some of the content in this unit was introduced in SI. The teacher is therefore encouraged to prod the student to participate fully in the class discussions and to research more about the various concepts presented here.

Suggested teaching/ learning activities

To effectively deliver this content in this unit, the teacher is expected to use group discussions, research, demonstration and assessment methods beyond those provided in the student's book.

Definition of a computer and computer science

(1 period)

This topic provides concise definitions of the terms computer and computer science.

Information to the teacher

The definitions provided in the student's book are generic obtained from internet sources such as Oxford online dictionary. It is important to divide the class into groups and guide them how to research on reliable sites to get alternative definitions of the two terms. To reinforce understanding of basic concepts, let the groups attempt activity I.1 on page 1 of the student's book.

Preparation

Carry with you pictures of various computers and peripheral devices.

Materials: computer, whiteboard, projector, student book and internet.

Teaching guidelines 1.1

- When dealing with this topic, impress on the students the fact that the computing machine was first born through inquisitive minds who wanted to invent through research and innovation. This should create a link between computer science and computer.
- Give the students a chance to present in class the definition of a computer and computer science.
- Let the student do learning activities 1.2 and 1.3 on page 2 of the student's book.
- Cover the relevant content on characteristics of computers after giving students a chance to mention common characteristics of computers that they know.

Characteristics of computers

(1 period)

This topic explains some of the characteristics of computers in terms of speed, accuracy, versatility, diligence, reliability and memory.

Information to the teacher

The characteristics provided in the student's book are not exhaustive. It would be a good idea to divide the

class into groups and guide them how to research on other characteristics of computers. To reinforce understanding of basic concepts, let the groups attempt activity 1.3 on page 2 of the student's book.

Materials: computer, wall chart, projector and internet.

Preparation

Carry a chart to class that lists the various characteristics of computers.

Teaching guidelines 1.2

- Guide the students to do Activity 1.3 on page 2 by suggesting relevant websites that have the characteristics of computers.
- Allow a brief brainstorming session for the students to discuss the characteristics that they find out.
- You can now cover the content in this section. Use the wall chart and the material in the student book to make it interesting.

Classification of computers

(6 periods)

This topic is based classifies computers according to size and processing power, function and type of data processed.

Information to teacher

In this section is divided into three sub-sections that classifies computers according to size and power, function and type of data they process. It is important to note that this classification is not con-

crete. This classification is merely based on the syllabus hence should not be considered as the standard method of classifying computers. Guide the students on researching other methods used to classify computers. For example, they may find out that computers may also be classified general purpose and special purpose.

Types of computers according to size and power

By the end of this section, the learner should be able to classify types of computers according to size and processing power.

Materials: student text book, wall charts, whiteboard, projector, computer connected to internet.

Preparation

Carry the pictures of supercomputers (both of old and current), mainframes, minicomputers and personal computers.

Teaching guidelines 1.2

- Using practical examples or illustrations, guide the students in identifying various types of computers according to size and processing power. This would help them carry out activities 1.4 to 1.8 from page 3 to 6 in the student's book.
- In most of these activities, you may divide the class into groups and give each student a chance to present findings from group discussions.

Types of computers according to function

By the end of this section, the learner should be able to classify types of computers according to function they perform.

Materials A computer connected to the internet, projector, video clip.

Preparation

Look for a video of robots at work. Also, a mobile phone is a special purpose computer – for communication..

Teaching guidelines 1.3

- Using practical examples or illustrations, guide the students in identifying three types of computers according to function they perform. These are servers, workstations and embedded computers. This would help them carry out in the student's book.
- In To help the students identify the three types of computers, you may be required to organize for a fieldwork visit to an organization or industry where servers, workstations and embedded computers are accessible.

Types of computer according to data type

By the end of this section, the learner should be able to classify computers according to data type.

Materials a computer connected to the internet, student text book, whiteboard, projector.

Preparation

Carry a chart to class that classifies computers according to the type of data they process.

Teaching guidelines 1.4

- Start by mentioning various types of data that exist in the world today e.g. temperature, pressure, speed etc. Yet we need computers to capture and process this data.
- You can now introduce the concepts of analog, digital and hybrid computers. Be careful to give real life examples e.g. how come at the petrol station the pump displays the volume of fuel sold (analog data) using digital displays.

Evaluate the learners understanding of the concepts presented by using a question and answer session.

Role of computers in the society

(3 periods)

By the end of this section, the learner should be able to appreciate the role of computers in the society.

Information to teacher

This topic is intended to motivate the students in appreciating importance of computers and other ICT devices in all sectors of our society. It would be good to deliver this topic by organizing field visits to the industry, business organizations,

health institutions, higher institutions of learning and government offices. This not only motivates the students liking of the subject but also form a strong career foundation motivates a student education.

Materials a computer connected to the internet, student text book, whiteboard, projector, video.

Teaching guidelines 1.5

- This topic requires a lot of group work discussions. Divide the class into groups and assign them application areas to research on. Let each group identify actual places such as airports, supermarkets, government offices and hospitals.
- Let students present their finding work in class. After that, you can cover the content in this section.
- Apart from areas discussed in the book, let each group identify other computer application areas through research and discussions.
- Before closing the topic, evaluate students understanding of the concepts presented by carrying out activity 1.11 of the student's book.

History of Computers

(5 periods)

By the end of this section, the learner should be able to elucidate the historical evolution of computers.

Information to teacher

This topic discusses different computer generations, technology used in each generation. Given that the technologies discussed are concepts from physics or electronics, it would be good idea to organize for physics practical sessions to demonstrate how vacuum tube, transistors and integrated circuits works. If possible, you can also organize for a visit to computer garage or museum where some of these computers are in display.

Materials a computer connected to the internet, student text book, whiteboard, projector, video.

Teaching guidelines 1.6

- To effectively deliver this topic, start by forming group discussion with responsibility of researching on evolution of electronic computers. Let the students also perform activities 1.12 - 1.16 in the student book. Evaluate the learners understanding of the concepts presented by using a question and answer session.

Answers for Unit 1

Assessment Test 1

1. It does not get tired - It can be programmed to perform a task repetitively - It does not have the human body weaknesses - Once programmed it follows instructions exactly without fail
2. (a) Mainframe: smaller than

supercomputer with slightly less processing power. Used as central computer or server on large networks that require centralised data processing. Minicomputer smaller than mainframe. Used by smaller organisations as central server/ also used for specialised applications e.g. the autopilot on aeroplanes.

- (b) Analog computer processes continuous data (analog data); Digital computer processes discrete data.
- (c) Server provides services to other computers on the network; Workstation used by the user, has limited processing power and requests services from the server.

Guideline to Activities

Activity 1.1: Definition and parts of a computer (page 1)

This activity requires the students to use a search engine to get appropriate definitions. For using Google we got the following definitions from online oxford dictionary:

Computer—An electronic device which is capable of receiving information (data) in a particular form and of performing a sequence of operations in accordance with a predetermined but variable set of procedural instructions (program) to produce a result in the form of information or signals.

Computer science-The study of the principles and use of computers.

This activity requires the student to identify and define the parts labeled as follows:

a: System unit – houses the CPU, fixed storage devices and other electronic components

b: Monitor or display unit used as visual display of data or information from the computer

c: Mouse - pointing devices used for controlling cursor on the screen

d: keyboard – input device used for entering data and instruction

Activity 1.2: Computer Science (page 2)

Both questions in this activity requires the learners to explore the domains of study offered in most Computer Science and Information technology programs especially at the university level. This includes Information systems, Artificial Intelligence, Distributed Systems, data communication and networks, security, database systems, human computer interaction, vision and graphics, numerical analysis, programming, software engineering, health informatics, bioinformatics and computational theories

Activity 1.3: Characteristics of a Computer (page 2)

This is an exploratory activity that requires students to brainstorm some of the characteristics of a computer that are later discussed on pages 2 and 3.

Activity 1.4: Classification of computer (page 3)

This is an exploratory activity that requires students to brainstorm some of the characteristics of a computer that are later discussed on pages 2 and 3.

Learners are expected to identify other factors that may be used to classify computers such as functions, shape, speed, cost, type of use.

Activity 1.5: Uses of supercomputers (page 4)

The learners are required to identify application areas of supercomputer such as Pleiades in high-fidelity physical simulations, study of spacecraft and aerodynamic forces and the detonation of nuclear weapons.

Activity 1.6: Mainframe computers (page 4)

The learners are required to identify application areas of mainframe computers such as System Z. For example, ATM, airline ticketing, credit card verification may be accessing mainframe computers to provide efficient transactions.

Activity 1.7: Distinction between mainframe and minicomputers

(page 5)

Learners are expected to draw comparisons between mainframe and mini computers in terms of physical size, speed, cost, and uses.

Activity 1.8: Types and uses of Microcomputers (page 6)

This activity requires the learners to walk around the school or its environments to identify various PCs that are in use such as desktop PCs, laptops, tablets and even smart phones.

Learners should be able to conceptualize the development of computers in terms of microprocessors that came into being in the 4th generation.

This activity requires the learners to identify the two types of computers namely: laptop PC and tablet.

Activity 1.9: Classification of Computers (page 8)

This activity requires the learners to walk around the school environment and identify various types of computers. In most cases, they may come across the school servers, workstations in offices, tablets and possible embedded computers.

This activity requires the learner to give advantages and disadvantage supercomputers and microcomputers in terms of processing speed, multi-user access, cost, portability, scalability, efficiency and reliability.

Activity 1.10: Classification of

Computers (page 9)

The two questions in this activity are exploratory because they requires the learners to research on the internet the three types of computers according to the type of data they process. It is from the research outcome that they are able to give advantages and disadvantages of each type in terms of power consumption, size, cost, simplicity of design, and processing power.

Activity 1.11: Role of Computers in Society (page 11)

This activity requires the learners to match the computer application areas marked 1,2...8 with applications on the right marked A,B..H on the right as follows: 1 => H; 2=>G; 3 none; 4=>D; 5=>B; 6=E; 7=F; 8=A. Note that stock control may also be assigned to supermarket but there is no match to the Bank on the left.

This is an open activity that requires learners to brainstorm on impact use of computers and other ICT devices in promoting peace and reconciliation, environmental protection and life skills in Rwanda.

This is a fieldwork activity that requires the teacher to organize for field excursion for learners to appreciate use of computers various sectors. It important to let the students visit institutions of higher learning, research centres, hospitals, police stations, banks and government offices where

computers are used to automate various tasks.

Activity 1.12: First Generation Computers (page 12)

This activity requires learners to research on the internet or explains from the student book reasons the 1st generation computers were large, consumed a lot of power and emitted lots of heat. One of the expected reactions is that vacuum tubes mostly rely on thermionic emission of electrons from a hot filament or a cathode heated by the filament hence need for excess heat emission.

Activity 1.13: Second Generation Computers (page 13)

This activity requires the learners to research on the internet or explain from the student book more examples of second generation computers apart from IBM's I401 and 7070, UNIVAC 1107, ATLAS LEO Mark III and Honeywell H200.

Activity 1.14: Third Generation Computers (page 13)

This activity requires learners to research on the internet or explains from the student book more examples of third generation computers apart from IBM 360 and ICL 19000 series.

Activity 1.15: Fifth Generation Computers (page 14)

This activity requires learners to research on the internet or explains from the student book more examples

of fifth generation computers.

Activity 1.16: Computer Generations (page 15)

This activity requires the learners to match computer generations 1,2, 3 and 4 with technologies used on the right marked A,B, C and D on the right as follows: 1 => B; 2=>C; 3 =>D; 4=>A.

This activity require the student to explore beyond what is provided in the student book to trace the development of electronic computers starting with the first generation to the present and future generation commonly regarded as the fifth generation.

1. First generation computers were large in size, consumed a lot of power, and produced a lot of heat.
2. The diagram should depict first generation (vacuum tubes)--> second generation (transistor)--> third generation (ICs)--> Fourth generation(microprocessor) -->fifth generation(Artificial intelligence)
3. Form of intelligence exhibited by machines or software to mimic human or animal behaviour.
4. VLIC contributed to development of the microprocessor
5. Artificial intelligence - Graphical User Interfaces -Small physical size but large processing power Easy programming

Computer System and Maintenance

Computer Maintenance

Computer Architecture and Assembly

Unit 2

Student's Book page 16 – 53. (18 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

1. Identify computer components and their functions(input, output, processing and storage)
2. Assemble and disassemble a computer, do minor maintenance

Learning objectives

Knowledge and understanding

1. Identify computer system, components and function of each component.
2. Identify computer peripherals , ports and connectors.
3. Differentiate types and purpose of computer case.
4. Identify different components inside computer case and their functions.

Skills

1. Attach and de- attach computer peripherals.
2. Clean internal and external computer components and protect the environment from contamination.
3. Compare different computer case form factors.

4. Identify elements inside computer cases.

Attitudes and values

1. the evolution and the importance of a computer in:-
2. Appreciate the guideline of attaching and disconnecting each component of a desktop computer properly.

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of

socio-cultural differences hence supports interpersonal respect and promotes national unity.

Communication: presentations in class made after group work will develop this skill very well.

Links to other subjects: Physics (Electronics)

Cross cutting issues addressed in this unit

- Inclusive Education: through group work, the needs of challenged students can be addressed more easily.
- Peace, gender and values education
- Financial education

Assessment criteria

Students can identify the internal components of a computer and properly fix them on right place. They can diagnose the cause of problem associated with each component.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

Instead of using live computers to teach this unit, consider getting scrap (computers that are no longer in use) to allow students the freedom to freely disassemble and assemble without fear. After they have known how to do it, they can now work with one or two live specimen provided.

Suggested teaching/ learning activities

To effectively deliver this content in this unit, the teacher is expected to use group discussions, research, practical demonstration and assessment methods beyond those provided in the student's book.

Computer system

(1 period)

By the end of this topic, the learner should be able to identify and explain the four components of a computer system. These are hardware, software, data and user.

Information to the teacher

Before, you introduce this topic, you need to define the term system. Let the students appreciate the important of a system such as the human body, school and political systems. It is important to mention that, in some references, a computer system is considered as comprising of three components: **hardware, software and liveware**. Make sure you disambiguate differences that may be raised by the students out of research from other references such as internet.

In reference to Fig. 2.1, divide the class into groups and guide each group through activity 2.1 in the Student's book.

Materials: computer, whiteboard, projector, student book.

Preparation

Be ready to compare other systems found in the world with the computer system.

Additional information teacher

- Guide the students to do learning activity 2.1 on page 17.

Teaching guidelines 2.1

- Ask the students to give some characteristics of systems emphasizing on how many parts work together to achieve a whole.
- You may as well draw the student's attention to Fig. 2.1 on page 17 of the student's book and use it to justify why a computer is regarded as a system when in use.
- You can now cover the content in this section.

Computer functions

(1 Period)

By the end of this topic, the student should be able to describe the four functions of a computer system: input, process, output and storage.

Information to the teacher

This topic introduces the students to the four basic functions of a computer. These are input, processing, output and storage. When introducing this topic, you need to demonstrate how a computer processes

data to information. Let the students appreciate the important of hardware, software and user in data processing.

Materials: Whiteboard, student text-book, protractor, computer connected to the internet.

Teaching guidelines

- To effectively deliver this topic, it is advisable to divide the class into groups. Let each group use Fig 2.2 on page 18 to identify the functional elements of a computer.
- The finding from each group should be presented to class for clarification on key concepts and configuration of a typical microcomputer.

Computer Hardware: Peripheral devices

(1 period)

By the end of this topic, the student should be able to identify computer peripherals devices in the computer lab.

Information to teacher

This topic requires practical approach to introduce the students to various devices connected to the computer's system unit. Let the students attempt to classify peripheral devices in the lab into input, output and storage devices.

Materials: student text book, wall charts, whiteboard, projector, movie,

computer connected to internet, sample peripheral devices.

Preparation

Download images or video clip showing external hardware components of a computer also referred to as peripheral devices. Let each student also identify the components physically in the computer lab or around the school environment.

Teaching guidelines 2.2

- Draw the attention of the learner to Figure 2.3 on page 19 in the student book. Let them do activity 2.2 on pages 18 and 19 to demonstrate understanding of the concepts covered in this section.
- It is advisable to cover the content in this section after this activity is done satisfactorily by all the students.

Computer Hardware: System unit

(1 period)

By the end of this section, the learner should be able to describe the system unit and its importance.

Information to teacher

This topic requires practical approach to introduce the students to explain the importance of the system unit found in most desktop computers. Let the students identify the two form factors (types) of system unit that comes with desktop computers namely tower type and desktop system unit.

Materials: A computer connected to the internet, projector, video clip.

Preparation

This is best taught when students are in computer lab.

Teaching guidelines 2.3

- Demystify the system unit for the learners. Let them know it as the home of the motherboard and processor.
- Also, show the class the ports on the motherboard.

Ports and connectors

(4 periods)

By the end of this section, the learner should be able to identify different ports and connections on the system unit.

Materials: a computer system unit, various peripheral devices, connector cables.

Teaching guidelines 2.4

- For all this section (it may take more than one periods) start by asking the students to do activities 2.4 on page 20 before covering the content and activities from pages 21 to 26.
- Evaluate the learners understanding of the concepts presented by using group discussions, activities and question and answer methods. Let each student demonstrate understanding of ports and

connectors required to connect peripheral devices to a computer.

Internal computer Components

(2 periods)

By the end of this section, the learner should be able to identify internal computer components.

Information to teacher

This topic introduces the students to the various components housed inside the system unit of a typical desktop computer. Due to the sensitivity of most of these components, it is advisable to sensitize the students on the effect of electrostatic discharge. During demonstration intended to practically identify the, let the students observe caution when handling the components. Let the students appreciate the important of UPS in protecting these delicate components.

Materials: a computer connected to the internet, student text book, whiteboard, projector, video, manufacturers manual.

Teaching guidelines 2.4

- Guide the student through activity 2.13 before covering the content in this section.
- Open the system unit for the student to be able to see the internal components.
- Through a class discussion and practical demonstrations, guide the learners to understand the role of the power supply unit and how it is connected to the system unit.
- Open the system unit case and let the learners observe the motherboard and identify various components attached on it.
- Help the learners identify the CPU on the motherboard. Through a class discussion and with the help of a diagram of the functional elements of the CPU and activity 2.15, help the learners to understand the role of the Control Unit and ALU.
- Ask the learners to do Activity 2.16 and guide them to differentiate between permanent (long-term) memory and short term memory concepts as in relation to computer memory. Guide them through activity 2.17 and help them to understand the role of primary memories (i.e. ROM and RAM) in a computer.
- By guiding learners through activities 2.18, 2.19, and 2.21, help the learners understand the roles of special memories i.e. cache, buffers and registers.
- By guiding learners through activities 2.21, 2.22 and 2.23 help the learners identify and appreciate the features, working and look for various secondary memory devices including optical storage media and solid state storage media.
- Carefully open a hard disk drive and guide the learners observe and understand its features and their functioning including disk plates,

spindle and read/write head. Guide them to understand the role of HDD in a computer.

- Guide the learners to identify and understand the role of various disk drives in a computer and laptop. These drives include hard disk drive, DVD/CD drive and USB drive.
- Demonstrate to the students how they can use the manufacturer's manual to identify components and configure them.

Evaluate the learners understanding of the concepts presented by asking relevant questions.

Assembling computers

(3 periods)

By the end of this section, the learner should be able to assemble computers when provided with various components.

Information to teacher

This topic requires practical approach to take the students through step-by-step process of assembling a desktop computer. Due to the sensitivity of most of internal components, it is advisable to sensitize the students to observe safety precautions to protect themselves and computer components from possible risks. remind them to always discharge static electricity that may have build on their body before carrying out the task of assembling a computer. During practical demonstration intended to

practically mount internal components like processor, memory or adapter card, students should observe care in order to avoid damaging the socket or connectors.

Materials: components of a disassembled computer, manufacturers manual.

Teaching guidelines 2.5

- Guide the student through activities 2.25 to 2.35.
- At each stage make sure that the correct safety precautions are taken by the student to protect themselves and the computer components.

There is need to evaluate the learners understanding and skills by giving each student an opportunity to demonstrate how to fix a device inside the system unit.

Disassembling Desktop computers

(3 periods)

By the end of this section, the learner should be able to disassemble a desktop computer into separate internal and external components.

Information to teacher

This topic requires practical approach to take the students through the reverse process of disassembling a desktop computer. Due to the sensitivity of most of internal components, it is advisable to sensitize the students to observe safety precautions to protect themselves and

computer components from possible risks. Remind them to always discharge static electricity that may have build on their body before dismantling a computer. During practical demonstration intended to practically detach internal components like processor, memory or adapter card, students should observe care in order to avoid short circuit that may harm them or damage the components.

Materials: desktop computer repair toolkit, manufacturer's manual.

Teaching guidelines 2.5

- Guide the student through the process of dismantling a desktop computer as demonstrated of the student's book.
- At each stage make sure that the correct safety precautions are taken by the student to protect themselves and the computer components.

Cleaning and disposal of computer components

(2 periods)

By the end of this section, the learner should be able to clean computer components.

Information to teacher

This topic requires practical approach to take the students through preventive maintenance of computer components. You may be required to demonstrate to the learners how to use tools such as vacuum blower, solvents and computer repair tools. Due to risks that may arise

from handling chemicals and electrical devices, sensitize the students to observe safety precautions to protect themselves from health risk or damage to hardware components.

Materials: brushes, blower, vacuum cleaner, soft towel, non-water based detergent

Teaching guidelines 6.4

- To test the student's understanding of the concepts and skills gained, guide them through activity 2.37 on page 51 to activity 2.39.
- At each stage make sure the students follow safety guidelines to protect themselves and computer components.
- Evaluate each student's understanding of key concepts and competence in performing preventive skills on computers and peripheral devices.

Answers for Unit 2

Assessment exercise 2.1 (page 41)

- l. (a) AC- stands for alternating current. This power keeps on changing direction. DC:- stands for direct current. It moves in one direction only.
- (b) Bluetooth:- a radio frequency signal used to set up hotspots within a radius of 300 m. It can penetrate walls. Infrared: sets up connectivity over a short distance. Bounces of surfaces and requires line of sight to

function.

- (c) Firewire is a faster more superior USB technology
 - (d) Five pin DIN: uses D-shaped ports with five pins to send receive data; P/S2 for mouse and keyboard connections uses six pin circular port and connector.
2. (i) USB (ii) Firewire (iii) COM port
 3. Smaller size, longer distance connectivity; less cumbersome; cheaper.
 4. You should have a computer with two VGA ports; Or one VGA port and a HDMI port so that you can connect one device to the VGA and the other to the HDMI. Alternatively if the computer does not have two ports, you need to buy a VGA adapter that matches the ports on your devices. After you connect the devices:
 - (i) Click the Start button, choose Settings, and click the System icon.
 - (ii) When the System page appears, click the words Advanced Display Settings in the screen's bottom-right corner.

Assessment exercise 2.2 (page 50)

1. (a) Integrated Drive Electronics (EIDE) and Serial Advanced Technology Attachment (SATA) are both interfaces for connecting the hard disk to the motherboard but SATA is

more recent and it supports hot swapping technology.

- (b) AT is a very old standard that was created by IBM for their own computers. ATX was developed by Intel to address some of the shortcoming of the AT standard that makes it unsuitable for the varied demands of personal computing.

AT boards were much bigger compared to ATX boards. This results in some drives overlapping the boards inside the case which meant that in order to replace the board, you would need to take everything out. This is very inconvenient for personal computers, thus the ATX boards were made narrower by almost 4 inches to ensure that there is no overlap.

- (c) PGA - Pin Grid Array is used in Pentium 3 computers; SECC - Single Edge Contact Cartridge used in Pentium 2 computers.
- (d) AMR (Audio Modem Riser), CNR (Communications and Network Riser) and ACR (Advanced Communications Riser) are slots that you can find on your motherboard that have the same goal: to install HSP (Host Signal Processing) devices to your PC. These devices can be modems, sound cards and network cards.

2. For quick working with minimum

effort; To avoid spoiling the place or component you are trying to repair.

3. - extended extractor; - antistatic wrist member; - Torx screw; driver - multimeter
4. -Industry Standard Architecture (ISA), Extended ISA (EISA), Peripheral Component Interconnect (PCI); Accelerated Graphics Ports (AGP); Video Electronics Standards Association (VESA); Audio Modem Riser (AMR); and Communication Network Riser (CNR).
6. Check the fuse. Also check whether the power cable running from the power supply is plugged onto the motherboard port or slot.
7. (i) Identify the manufacturer of the BIOS chip.
(ii) Back up the CMOS Settings and restart the computer using a combination of CTRL + ALT + DELETE keys.
(iii) Enter the CMOS settings program using the specified key or combination of keys, and then write down the settings.
(iv) Backup the old BIOS in case the upgrade results to system failure.
(v) Insert the manufacturer's BIOS utility disk. The disk contains a program that automatically flashes the BIOS.
(vi) Restart the computer. If successfully done, the BIOS

retains the new firmware.

Assessment exercise 2.3(page 53)

1. A UPS to protect his/ her electronics.
2. - worn out feeder wheels; - low quality paper or paper; - Creased paper
3. - Misaligned / dirty print heads; - Lack of appropriate print driver; - lack of print toner / low quality cartridge or toner.
4. Toner drum; or cartridges paper tray; feeder rollers.
5. To reduce wear and tare due to friction; to reduce crashing of sensitive devices like the hard disk.
6. CD/DVD: use the cleaner CD and solvent; Keyboard: use blower, special brushes and vacuum cleaner; Monitor: use blower, smooth absorbent cloth.

Guideline to Activities

Activity 2.1: Computer Components (page 17)

This activity requires learners to identify and explain function of the four components of a computer system namely: hardware, software (programs), user and data.

Activity 2.2: Peripheral Devices (page 18)

This activity requires learners to physically identify various input, output and storage devices attached to a computer. This may include keyboard,

mouse, joystick, speakers, microphones, printer, scanner, external storage media and webcam discussed later in the section.

Activity 2.3: System Unit (page 20)

The three question of this activity requires learners to identify and gain understanding on systems unit, form factor (tower and desktop) and design issues (office space vs. elegance) of the system units that are typical to desktop computers.

Activity 2.4: Ports and Connector (page 20)

This activity requires the teacher to assist the learners physically identify various ports at the back of the systems unit. This is intended to assist the students fill the port labeled A to E as follows: A=> PS/2 ports; B=>serial port; C =>Firewire port; D=>USB ports; E=>parallel port; F=>audio jacks/port.

Activity 2.5: Serial Connector (page 21)

The three question of this activity requires learners to identify whether the computers in the school environment have serial port. Most of the modern PCs have phased out the port in favour of USB, HDMI and Firewire port hence the student may not physical find it on laptops and most desktop PCs. The teacher should assist the student appreciate reasons the port has been phased out.

Activity 2.6: USB Port and Connector (page 22)

This activity requires the learners to appreciate the popularity of USB as the de facto port that has replaces all the other serial and parallel ports. The activity also requires the teacher to assist the learners to physically identify and practically connect devices to USB ports.

Activity 2.7: PS/2 Port and Connector (page 23)

This activity requires the learners to identify PS/2 port using colour code or symbols on older types of desktop computers. Given that the ports have been replaced by USB ports on modern PCs, the teacher may use other alternatives such as YouTube because it may not be practically possible to identify and connect devices to PS/2 ports.

Activity 2.8: VGA Port and Connector (page 23)

This activity requires the learners to identify a VGA port at the back of the PC and give the count of pins as 15. Given that the VGA has been replaced by HDMI in some devices especially on laptops, the teacher may physically demonstrate how to connect a projector or monitor to the port or use alternatives such as YouTube

Activity 2.9: Audio Port and Connector (page 24)

This activity requires the learners to sound (out) and microphone port using colour codes or symbols on system

unit or laptop. The teacher should also assist the learners on how to connect headphones, microphone or external speakers to the appropriate sound ports.

Activity 2.10: Network Interface (page 24)

To draw a clear distinction between a network adapter and onboard modem, the teacher should assist the student to distinguish between the RJ45 and RJ11 ports. You may also demonstrate practically how they differ in terms of data transmission and placement on the motherboard.

Activity 2.11: HDMI Port and Connector (page 25)

This activity requires the learners to physically identify HDMI port on devices such as laptops, desktop computers and other multimedia devices such as TVs and projector. Because HDMI and USB port almost resembles each other, it would be good to demonstrate the difference in terms of connector and number of pins on the port. Let the students practically connect HDMI-enabled devices to a computer.

Activity 2.12: Connecting Peripheral Devices (page 26)

This is a practical activity that requires the learners to identify SCSI port on devices such old models of desktop computers or from alternative sources. The learners are also expected to demonstrate how to connect peripheral devices to appropriate ports.

Activity 2.13: Internal Computer Components (page 27)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners disassemble a computer and open the system unit casing. This procedure should be handled with care to avoid electric shock or damage to internal components.

Activities 2.14 to 2.24 power supply unit, CPU and computer memory. (pages 30-39)

- These are largely general knowledge activities that are meant to help the learners understand the functions of various internal computer components with the help of their real life knowledge.
- For such activities, give the learner very clear instructions so that they acquire the intended knowledge. Ensure you relate the results of activities with the functions of the respective part of computer under discussion.
- For the activities that require observation, avail the components and guide the learners to clearly identify the features.

Activity 2.25: Assembling a Computer (page 42)

This is a practical activity that requires the learners to identify various tools such as extended extractor, antistatic wrist member, screw drivers and Ethernet crimping tools in a computer repair toolbox.

Activity 2.26: Mounting a Hard

Disk (page 43)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners mount SATA or IDE hard disk drive in the system unit. This procedure should be handled with care to avoid electric shock or damaging the hard disk drive.

Activity 2.27: Installing Optical Drives (page 43)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners mount SATA or IDE optical drive (CD/DVD) in the system unit. This procedure should be handled with care to avoid electric shock or damaging the optical drive.

Activity 2.28: Mounting a motherboard (page 44)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners mount a motherboard in the system unit. This procedure should be handled with care to avoid electric shock or damaging the motherboard.

Activity 2.29: Installing a Computer Memory (page 45)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners mount a motherboard in the system unit. This procedure should be handled with care to avoid damaging the RAM modules or the motherboard.

Activity 2.30: CMOS Battery Replacement (page 45)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners identify and replace CMOS battery mounted on a motherboard. This procedure should be handled with care to avoid electric shock or damaging the motherboard.

Activity 2.31: BIOS Upgrade (page 46)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners to backup and upgrade BIOS firmware. Because BIOS comes with manufacturer's instructions, BIOS software-based upgrade is a delicate procedure that should be handled with care to avoid crashing the firmware.

Activity 2.32: Adapter Card (page 46)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners identify functions of network interface card shown in Fig. 2.34 demonstrate how to mount it on the motherboard. This procedure should be handled with care to avoid electric shock or damaging the adapter card.

Activity 2.33: Installing a Microprocessor (page 47)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners

mount an Intel or AMD processor on the motherboard. This procedure should be handled with care to avoid electric shock or damaging the microprocessor contact or the socket on the motherboard.

Activity 2.34: Laptop Battery Replacement (page 48)

Unlike desktop computers that receive direct power from mains outlet, laptops have a rechargeable battery that makes them. It is important to give students a practical demo on how to replace laptop battery. In case there is no laptops that can be used to accomplish this activity, the teacher may use alternative sources such as YouTube or multimedia content to simulate the activity.

Activity 2.35: Laptop Memory Upgrade (page 48)

Like desktop computers, you can replace or upgrade RAM of a notebook PC. It is important to give students a practical demo on how to replace laptop battery. In case there is no laptops that can be used to accomplish this activity, the teacher may use alternative sources such as YouTube or multimedia content to simulate the activity.

Activity 2.36: Assembling a Desktop Computer (page 50)

This is a practical activity that tests ability of the learners in assembling desktop computers. In a guided practical session,

let each student demonstrate how to assemble a computer from scratch. This includes mounting motherboard, processor, memory modules and storage drives.

Activity 2.37: Cleaning Computer Devices (page 51)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners perform basic maintenance services using solvents to clean computer accessories. Students should be introduced to various cleaning solvents and their effect to their health and environment.

Activity 2.38: Blowing Dust and Debris (page 51)

This is a practical activity that requires the guidance of the teacher or qualified lab technician to assist the learners perform basic maintenance services using hand-held vacuum cleaners and dust blowers. Students should be advised on how to protect themselves from effect of dust and debris from the computer.

Activity 2.39: Safety Precautions (page 52)

This is activity is intended to nurture patriotism in the process of how the government programme of cleaning and

beautification of the country relate to e-waste. Let the students participate in cleaning exercise especially in disposal of electronic wastes in the school and the surrounding environment.

Unit Test 2 (page 53)

1. USB: Universal Serial Bus; SCSI: Small Computer System Interface
3. USB; GGA; HDMI
4. CRT: uses electron gun technology. Advantage: 1. Cheaper to produce; LCD: uses liquid crystal display technology; Advantages: 1. does not flicker; 2. angular view remains the same; 3. High resolution pictures and text; 4. Consumes less power.
5. Powerful processor - duo core 2Ghz and above; Large RAM 1 GB and above; Large hard drive 100GB and above;; Multimedia speakers.
9. Check the analog cable/ digital cable that connects the CDROM player to the motherboard whether it is properly plugged in; Check whether the speakers are connected correctly to the jack ports; Check whether the speakers are powered on.
10. It keeps the computer working at optimum level and prevents unnecessary breakdowns caused by dirt and water. You can do the following preventive maintenance:
 - (i) Use blower to blow out dust
 - (ii) Wipe the computer clean with a soft absorbent piece of cloth and non-water based cleaner liquid.

Computer System and Maintenance

Computer Maintenance

Unit 3

Safe and Ethical Use of Computers

Student's Book page 54 – 61. (12 Periods)

Key unit competence

By the end of this unit, the learner should be able to integrate safety guidelines, ergonomics and ethical issues to have a good working environment.

Learning objectives

Knowledge and understanding

1. Identify safety procedures at working place environment and procedures to protect the environment from contamination.
2. Identify procedure to protect equipment from physical damage
3. Identify Ergonomic Concept and associated Health Problem.
4. Identify Ethical Issues, Software Licensing, Anti-piracy, Forms of software license agreement (open source, proprietary, freeware and shareware)

Skills

1. Integrate general computer use safety procedures.
2. Able to protect equipment from physical damage.

Attitudes and values

1. Show the concern about to work in safe environment to protect computer against physical damage.
2. Identify ergonomic issues and health related risk due to improper use of computer.

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence supports interpersonal respect and promotes national unity.

Communication: presentations in class made after group work will develop this skill very well.

Links to other subjects

Physics (Electronics)

Cross cutting issues addressed in this unit

- Inclusive Education: through group work, the needs of challenged students can be addressed more easily.
- Peace and Values education.
- Financial education.

Assessment criteria

Can apply safety guidelines during computer use and repair, keeping health precautions, respecting legal and ethical issues.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

This unit prepares the learner to acquire basic computer safety and safe e-Waste disposal skills and knowledge.

Materials needed

Computers, Projector, UPS, Repair tool kit, Example of Some Commercial and Open-Source Software, Some samples of Ergonomics equipment like mouse, keyboard, screen, chair.

Suggested teaching/ learning activities

To nurture safe and ethical use of computers, the teacher is expected to use group discussions, illustrations, question and answer method, assessment questions beyond those provided in the student's book.

General Safety Guidelines

(2 periods)

By the end of this topic, the learner should be able to identify and follow laid down rules that govern use of computers in the workplace and show concern to protection of the environment.

Information to the teacher

This topic is aimed at sensitizing the students on risks associated with use of computers and other ICT devices. Though the student's book outlines general guidelines that need to put in place, it is the responsibility of the school or the computer teacher to formulate specific procedures that must be followed when handling computers and computer accessories. These include safety procedures in the computer labs, workplace, as well as procedures to protect the environment from e-waste

You may also guide the students in exploring other safety guidelines from the Internet to add on those provided in the student's book.

Materials: Computers, Projector, UPS, Repair tool kit, Example of Some Commercial and Open-Source Software, Some samples of ergonomics equipment like mouse, keyboard, screen and chair.

Preparation

The activities in this section are open discussion that requires discussion and brainstorming. Encourage the students to form discussion groups to perform each of the task in the activities.

Teaching guidelines 3.1

- This topic requires exposure to factors guidelines and procedure to be followed by students in the computer lab. Therefore, the teacher is required to use real life examples when give discussing health related problems while the learners find cause and solutions.
- It is advisable to refer the documents such those provided by REMA and WHO when formulating safety guidelines and procedures. Let the students appreciate each of the guidelines and procedures before undertaking activity 3.1 on page 55 of the student's book.

Ergonomic Furniture and Equipment

(2 periods)

By the end of this topic, the learner should be able to define ergonomic concept and identify health problems that are caused by poorly designed equipment.

Information to teacher

This topic is aimed at sensitizing the students on risks associated with poor use of computers and other ICT equipment. It is important to demonstrate to the students good sitting posture and guide ergonomic considerations when acquiring accessories such as keyboard, mouse, computer monitor and other equipment.

Apart from the content provided in the student's book, you may for group discussions to research on principles good ergonomics design, standard furniture and sitting posture.

Do not forget the need to consider users with physical challenges and or any other challenges and the need to take care of them.

Materials: student text book, wall charts, whiteboard, projector, movie, computer connected to internet.

Preparation

Download images or videos showing various ergonomic features of a good computer room.

Teaching guidelines 3.2

- Let the learner study the picture showing a smoke detector. Do you have such in your computer room?
- Cover the content in this section specifically mentioning what the first reaction of a person should be in case the smoke detector gives a warning of a fire breaks out

in the computer room.

- Finally, emphasise the need for a fire assembly point. Visit the fire assembly point. In case such a point does not exist in the school, the teacher to work hand in hand with the school administration for the class to have an outdoor activity one of the days to go and map out a location in the school that can be used as a fire assembly point.

Fire safety guidelines

(2 periods)

By the end of this section, the learner should be able to demonstrate understanding in fire safety guidelines.

Information to teacher

This topic is aimed at sensitizing the students on risks associated with fire outbreak in the school or computer lab. It is important to demonstrate to the students how to use a various types of fire extinguishers. The computer teacher may also organize for the fire response and assembly drills to let the students appreciate importance of fire extinguishers and smoke detectors.

Materials: A computer connected to the internet, projector, video clip, fire extinguisher.

Preparation

This is best taught when students are in the lab, and later in the field to demonstrate use of the fire extinguisher to out a small teacher controlled flame.

Teaching guidelines 3.3

- Guide the student in carrying out learning activity 3.3 on page 58 in the student's book.
- Let the learner study the picture showing a smoke detector. Do you have such in your computer room?
- Cover the content in this section specifically mentioning what the first reaction of a person should be in case the smoke detector gives a warning of a fire breaks out in the computer room.
- Finally, emphasise the need for a fire assembly point. Visit the fire assembly point. In case such a point does not exist in the school, the teacher to work hand in hand with the school administration for the class to have an outdoor activity one of the days to go and map out a location in the school that can be used as fire assembly point.

Climatic and Weather Change

(2 periods)

By the end of this section, the student should be able to identify procedure to protect equipment from climatic conditions.

Information to teacher

In this topic, you are required to

sensitize the students on climatic and weather changes that affect computers and computer accessories. Let the students form research group to discuss some of the possible causes of overheating, high humidity and low humidity on devices. This will help them appreciate need to use dehumidifiers and thermoregulators in some seasons or parts of the country.

Materials: website with content on climate and weather patterns in Rwanda, maps, projector, student book.

Teaching guidelines 3.4

- Let the learner study maps and pictures showing climatic conditions such as humidity and temperatures in various parts of Rwanda.
- Cover the content in this subsection by relating demonstrating effect of weather changes on computer components. For example, it is advisable to demonstrate how high humidity causes corrosion of metallic parts of hardware devices.
- At the end of the period, evaluate the learners understanding and skills by asking relevant formative questions.

Physical Damage

(2 period)

Before the end of this section, the learner should be able to identify procedure to protect equipment from physical damage.

Information to teacher

In this topic, the teacher should use practical examples to guide the students in identifying problems arise in their school computer lab which can cause physical damage. In each case, assist the individual students or groups to provide solutions to identified problems.

Materials: triboelectric objects, website with content on electrostatic discharge, projector, student book.

Teaching guidelines 3.5

- Let the learner practice on electrostatic discharge by performing an experiment in class or physics lab. This will help them answer questions in activity 3.4 on page 59 of the student's book.
- Let the student Students identify problems that may arise from physical damage in the computer lab. It is from their feedback that the you can suggest solutions to the problems
- At the end of the period, evaluate the learners understanding and skills by asking relevant formative questions.

Ethical Issues

(2 periods)

By the end of this section, the learner should be able to adhere to identify ethical Issues, software licensing, anti-piracy, and forms of software license agreement.

Information to teacher

In this topic, the teacher should use guide the students in a brainstorming session to identify ethical challenges that result from use of computers. In groups, let the students discuss on types of software license agreement (proprietary, freeware, shareware, and open source). It is from such discussions that the learner can identify ethical issues relating to software such as piracy.

Materials: policy documents on computer ethics, projector, student book and internet.

Teaching guidelines 3.6

- Let the learner identify ethical challenges arising from use of computers and mobile devices. This will help them answer questions in activity 3.6 on page 61 in the student's book.
- Let the students list some software and classify them under open source, commercial (proprietary), freeware and shareware. Note that more on forms of software licenses is covered in the next unit.
- At the end of the period, evaluate the learners understanding and skills by asking relevant questions.

Answers for Unit 3

Guideline to Activities

Activity 3.1: Safe Use of Computers (page 55)

1. This activity requires the learner

to identify factors such as poor sitting posture, prolonged use of computers, ventilation, poor display unit and electric shock.

2. Learners are expected to identify possible health risks such as electrocution or electric shock.
3. Drinks by drop on computer accessories causing short circuit or damage to electrical parts while solid food substances may cause clog moving parts or keys on the keyboard
4. This activity requires the learner to identify health risks caused by electromagnetic and radio waves relating to vision and hearing.

Activity 3.2: Computers and Environmental Protection (page 57)

1. This activity requires the learner to define the term e-waste (electronic waster) and how identify whether the government has laws or policy that govern disposal of such e-waste.
2. This activity requires the learner to use Internet to download the e-waste policy guideline from REMA website.
3. This activity requires the learner appreciate the role of REMA in environmental protection, especially implementation of laws and policies that govern waste disposal.
4. This activity requires the learner to use their creativity, problem solving and innovative approaches

to demonstrate how e-waste can be converted to commercial products.

Activity 3.3: Fire Safety Guidelines (page 58)

This activity is intended to help the students understand and appreciate the importance of installing fire water-based and non-water based fire extinguishers. Let the students participate in sample fire-fighting drills in that may be organized by the school from time to time.

Activity 3.4: Electrostatic Discharge (page 59)

1. This activity requires the teacher to demonstrate how electrostatic charges build up by conducting basic experiments using triboelectric objects.
2. This activity requires the learners to repeat the experiment of charging and discharging triboelectric objects to experience the effect of static electricity.
3. This activity is intended to help the students appreciate the importance of discharging static electricity that may have unknowingly that may damage electronic components such as silicon chips.

Activity 3.5: Power Protection Devices (page 60)

1. This activity requires the learner to practically connect a UPS to the mains outlet and the computer. This procedure should be handled with care to avoid electric shock

that may or damage to computers that result from poor power connection.

2. This activity requires learner to appreciate the importance of UPS in regulating power from mains supply. The UPS also serves as a backup in case of power outage hence allowing the user to shut down the computer using normal procedure.
3. This activity requires the learner to identify factors to be considered when acquiring UPS such as discharge rate, power rating, cost and size.

Activity 3.6: Ethical Issues (page 61)

This activity requires the learner to appreciate the impact of ICT on ethics, privacy and intellectual property. Let the student explore of various on various policy that safeguard or regulate effects of ICT on the three issues.

Unit Test 3 (page 61)

1. UPS, Backup generator
2. To avoid accidents like too much stumbling on each other or suffocation;
3. Switch off power from main switch or circuit breaker; Use non-water non-powder based extinguisher to put out the fire.
4. The food / drink may pour into sensitive computer parts causing clogging or rusting.

5. The fine powder may go into moving parts to cause friction hence wear and tear; may cause crashing of HDD
6. UPS smoothens the spikes, provide power during brownout or blackout; Surge protectors only sieve out the spikes and has no backup power when there is a brownout or blackout.
7. CVS: flickering monitor; Monitor too bright or shinny reflections on the screen.
8. Keyboard: a good layout to ensure easy finger reach from home keys to others; Furniture: good standard furniture that is comfortable and helps the user to maintain a straight back; Adjustable displays: to help user adjust for best posture and eye level.
9. Students warch on the internet ,the guidelines on waste managment in Rwanda.
10. Open source, proprietary (commercial), freeware and shareware.

Computer Maintenance

Unit 4 Computer Software Installation

Student's Book page 62 – 84. (16 Periods)

Key unit competence

By the end of this unit, the learner should be able to:-

- Install Operating System and other Application software
- Use disk management tools.

Learning objectives

Knowledge and understanding

1. Explain and differentiate types of computer software.
2. Identify form of software license agreement.
3. Identify system required to install any software.

Skills

1. Able to create partitions on hard disk and formatting a disk.
2. Use disk management tools to manage a disk.
3. Install some of system and application software.

Attitudes and values

Be aware of how to install a software and how to manage a disk.

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence supports interpersonal respect and promotes national unity.

Communication: presentations in class made after group work will develop this skill very well.

Links to other subjects: The syllabus does not explicitly document subject related to this unit.

However, the content of this unit indirectly links to Mathematics, Economics and Legal studies.

Cross cutting issues addressed in this unit

- Inclusive Education: through group work, the needs of challenged students can be addressed easily.
- Peace, gender and values education
- Financial education

Assessment criteria

Students can install and troubleshoot a software.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

This unit prepares the learner to acquire basic computer safety and safe e-Waste disposal skills and knowledge.

Materials needed

Computers, Projector, repair tool kit, Software, External Hard Disk, Network connection, Flash Disk, Windows Genuine installation DVD, Any Application software for installation

Suggested teaching/ learning activities

To effectively deliver this topic, the teacher is expected to use demonstration, research, group discussions and other assessment methods.

Classification of Computer Software

(1 period)

By the end of this topic, the learner should be able to explain and differentiate types of computer software.

Information to teacher

Before you introduce this topic, it is important that you research on different ways of classifying software. This because, the of method used to classify types of computer software may differ depending on context and the reference used. For this reason, we have discussed types of software by generally classifying them according to purpose and acquisition.

: Computer, Internet connectivity, software installation media.

Preparation

Make sure you have required hardware and resources. In case you don't have licensed copies of proprietary software you may demonstrate practical skills using Open-Source software. You can also request the school to pay for subsidized licensed software for students

Teaching guidelines 4.1

- To effectively nurture knowledge, skills and attitude required in this topic, you may also introduce the students to basic skills in using system and application software.
- Take the students through a group session to distinguish between

system software and application software. This will help them in carrying out activity 4.2 on page 65 of the student's book.

Software License

(1 period)

Information to teacher

The content of this topic is similar to the one covered in the previous unit on forms of software license agreement. Through group discussions, let the students review on forms of software license (proprietary, freeware, shareware, and open source). It is from such discussions that they can appreciate ethical considerations before proceeding to the section on software installations.

Materials: computer, Internet connectivity, software installation media and user guides.

Preparation

This topic can be confusing to the learner. make sure the learners distinguished the four types of software licences base on EULA. For example open-source license may not be freeware but may be shareware.

Teaching guidelines 4.2

- Using examples, let the students identify the four types (forms) of software licensing using. It is important to emphasize on the ethical use of each of the four categories in line with EULA.

Take the student through Activity 4.3 on page 66. Also use group

discussions to confirm students mastery of concepts taught during the period.

Disk preparation

(2 periods)

By the end of this section, the learner should be able to prepare a disk for software installation.

Information to teacher

This topic is aimed at introducing the students to two important but sensitive disk preparation procedures. These are disk partitioning and formatting. Due to sensitivity of the two processes, we recommend that the teacher carefully monitors the students as they carry out the task to avoid loss of data or programs. It is advisable to use external storage media or computers which do not hold sensitive information.

In regard to file systems, it is important to divide the class into groups to research on different file systems supported by Windows, Linux and MacOS operating systems.

: A working computer which has no operating system.

Preparation

This period should be very interesting to the learners and should be covered in a practical environment e.g. computer lab.

Teaching guidelines 4.3

- To effectively deliver skills in disk

management demonstrate how to partition and format.

- Measure the students competence by assessing their response to exercises activities 4.5 on page 68.

Disk Management

(1 period)

By the end of this section, the learner should be able to carry out disk management routine in Microsoft Windows or any other operating system.

Information to teacher

This topic is aimed at building on disk preparation covered earlier. Therefore, this topic requires practical approach to demonstrate how to carry out disk management procedures such as cleanup, scanning, running system file checker (sfc), disk defragmentation, disk compression, backup, and changing the boot sequence.

To avoid loss of data or programs, sensitize the students on safe disk management procedures and closely monitor practical sessions.

: student text book, data projector, video clip or pictures showing disk management progress.

Teaching guidelines 4.4

- Guide the student to go through the instructions in the student book to demonstrate each disk management routines.
- The students should also form groups

to carry out activities 4.6 on page 69.

- Use question and answer method to confirm whether the student has mastered the objective of this topic.

Installing Operating System

(4 periods)

By the end of this section, the learner should be able to install the operating system.

Information to teacher

Before you demonstrate how to install the operating systems, its is important to divide students into groups with each group having a computer and copy of the operating system to be installed. Ensure that each group has gone through the documentation that comes with the OS to identify minimum or recommended system requirements. Let the students practice how to change boot sequence settings before installing an operating system.

Teaching guidelines 4.5

- Guide the learner to follow the steps as detailed in the students text book in order to successfully install an operating system.
- Although the student book specifically refers to installation of Windows 10, this is only for demonstration purpose. You are required to demonstrate how to

install any other type of operating system including latest versions of Windows, Linux and Android.

- After installation, let the students go through activity 4.7 on page 80.

Installing Device Drivers

(1 periods)

By the end of this section, the learner should be able to install device drivers.

Information to teacher

Once the students have successfully installed the operating system, demonstrate how to install and configure device drivers using plug-and-play and manual installation. Sensitize the students on the need to read the manufacturer's manual to identify the right drivers to avoid corrupting an operating system. : computer, hardware device, drivers, manufacturer's manual and internet and student book.

Teaching guidelines 4.6

- Guide the student to go through the step-by-step process of installing device drivers. We highly recommend that you first demonstrate how to install drivers through plug-and-play and manually..
- To evaluate each student's competence, give them a supervised practical exercise and assess it by asking relevant questions.

Installing Application Software

(1 periods)

By the end of this section, the learner should be able to application software such as Microsoft Office 2013 on a typical desktop computer.

Information to teacher

Although the student's book discusses how to install Office 2013, you are required to demonstrate how to install any type of application software. This may be done after installation of an operating system not necessarily after device drivers installation. Once the students have successfully installed an application software, demonstrate how to launch and use them to perform specific tasks. Sensitize the students on ethical consideration to avoid installation of freeware or shareware that comes with malware and spyware that are harmful to a computer. It is advisable to first install an antivirus or internet security software before installing any application software:

Teaching guidelines 4.7

- Guide the student to go through the step-by-step process of installing an application software. We recommend that you first demonstrate how to install an App before the students install licensed Apps in groups.
- To evaluate each student's competence, give them a supervised practical exercise and assess it by asking relevant questions.

Answers for Unit 4

Assessment Exercise 4.1 (page 80)

1. Open source refers to software whose source code is available to users while proprietary software whose source code is hidden from users
2. This procedure requires use of Disk management utility. Right click This PC, then Manage to access the utility
3. Check the manufacturer's manual on minimum or recommended memory capacity; processor type, processor speed and hard disk size.
4. Piracy is illegal. Hence installing genuine software is an ethical requirements against piracy and intellectual property theft.

Guideline to Activities

Activity 4.1: Classification of Computer Software (page 62)

This activity requires the learner to research on the internet or before going through the classification provided in the student book. It is important to make the students understand that there are various ways of classifying software as they might discover as they explore various internet sources.

Activity 4.2: Classification of Software (page 65)

This is an exploratory activity that requires the learner to research on various means of acquiring software

such as download, buying from vendors or developing their own software.

Activity 4.3: Software License (page 66)

This activity requires the learner to practically identify system and application software installed on computers within the school or computer lab. The second part is meant to help the learner appreciate copyright and intellectual property rights needed to be protected in an end-user license agreement.

Activity 4.4: Software Installation Requirements (page 67)

This activity requires the learner to research from internet system requirements for Microsoft office, antivirus and Linux software.

Activity 4.5: Disk Management (page 68)

This activity requires the learner practically demonstrate various ways of accessing Disk management utility in Windows 10. For example, if This PC icon is available on the desktop, the user can display disk management window as follows:

1. Right click This PC icon,
2. Click on Manage to display Computer management window
3. Click Disk management under storage
4. Right click the drive you wish to manage on the right pane

Activity 4.6: Disk Cleanup (page 69)

This activity requires the learner practically perform disk cleanup procedure by following the steps provided in the student book or using cleanup in the search box to display Disk Cleanup utility on the startup menu.

Activity 4.7: Software Installation (page 80)

1. This activity takes care of installation of earlier or later versions of Microsoft Windows. It is important to demonstrate how to install Linux on top of Windows or as a stand-alone operating systems in a different partition
2. This activity lets the learner customize Windows 10 user look and feel such as the background theme, and icons displayed on the desktop. This is achieved by right clicking an empty space on the desktop then clicking Personalize from the shortcut menu

Activity 4.8: Device Drivers Installation (page 82)

This activity requires the students to install device drivers for common devices like printers scanners.

Activity 4.8: Device Drivers Installation (page 84)

This activity tests the student's skills on installation of new Apps such as

browsers, games and productivity software on Windows 10.

Unit Test 4 (page 84)

1. The help manual is important in identifying the hardware specifications required in order to install the new software.
 - It helps you to know the procedures to follow in order to install the new software.
2. Make sure you have the manufacturers disk or you download the relevant file.
 - Read manufacturers manual.
 - Double click the Setup.exe file and install the software as per instructions.
 - In case special configurations are needed, configure the software as per the requirements.
3.
 - The use needs
 - The operating system environment
 - The hardware requirements needed to run the software
4. Memory requirements; Processor speed; Hard disk size;
5. It helps prove that you rightfully own the software and it is genuine.

Computer System and Maintenance

Computer Arithmetic

Unit 5

Number Systems

Student's Book page 92 – 117. (18 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

1. Compute numbers in different base systems
2. Perform arithmetic operations on binary numbers

Learning objectives

Knowledge and understanding

Differentiate number base systems.

Explain the conversion and use of arithmetic operations in different base systems.

Skills

Convert a given positive number from one base to another.

Convert fractional numbers.

Convert a given negative decimal number to binary base.

Apply arithmetic operations.

Attitudes and values

Appreciate numbers base conversion and the use of arithmetic operators in binary base systems.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in four number systems. This include manipulating binary, decimal, octal and hexadecimal numbers using arithmetic operators.

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires group work or pair work as covered in the student book unit 2, the student gets a chance to nature skill

such as cooperation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of modern digital devices using binary logic. The learner is expected to transfer these skills to Boolean logic and digital electronics.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Critical thinking: Throughout this topic, we have emphasized on critical thinking in learning activities and assessment exercises.

Links to other subjects

This unit is highly linked to mathematics and physics: In mathematics, the unit is closely linked to arithmetic, and Boolean algebra. Understanding of digital electronics in physics is instrumental.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Environment and sustainability:** In first section of the unit, we have

addressed the issue of environmental (green energy) and sustainability of digital technologies.

3. **Financial Education:** This is indirectly implied in activity on benefits of digital technology that has resulted to cost-effective devices that are cheaper to make, buy, and maintain. Furthermore, digital devices save on power cost because they consume less energy.

Assessment criteria

In this unit, the students are required to correctly perform conversion and arithmetic operations in different numbers systems. Beside activities and exercises provided in the student book, the teacher should use various assessment methods and tools to test whether the learner has acquired necessary knowledge, skills and attitude in numerical computations.

Suggested teaching methodology

Guided discovery

Research

Question and answer

Discussion

Background information

The number systems unit introduces the learner to various concepts and manipulation of the numbers; with more emphasis on binary numbers. Apart from appreciating various number base systems, we make effort to connect such numbers to real world of mathematics and physics. This will help the learners

develop a strong foundation required in not only in computer science but also in engineering and digital electronics.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, research, question and answer method, group discussions and oral in delivery of this unit content.

Fundamentals of number systems

(2 periods)

This section introduces the learners to basic concepts associated with number systems. These include bit, nibble, byte, word, and magnitude (weight-by-position).

Information to the teacher

Fundamental concepts is meant to provide overview on origin of natural numbers before handling binary number systems used to represent data in digital computers. Note that this unit may be relatively to some students because it borrows a lot from mathematics and physics. To address this challenge, first introduce the learners to basic arithmetic using decimal number system. You can then introduce the other number systems gradually through demonstration using examples from the student book or any other relevant source.

Materials: Students book; website tutorial sites and other reference materials. You can also use wall charts depicting various

number base systems.

Preparation

Be prepared to handle weak students who have poor background in mathematics or physics. You may need to give such student to more specialized attention using practical exercises.

Teaching guidelines 5.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to study and write a brief report for Activity 5.1 that tests the student prior knowledge on decimal number system. Let the group leader give each member an opportunity to give a description as the secretary notes down the key points.
- The groups should present their findings in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners to understand the importance the ASCII code and binary logic digital electronics.

Additional information teacher

After introducing the learner to fundamental concepts, encourage them

to research how a computer represents data in electronic circuits, optical disks, magnetic disks and solid state storage devices.

Number Base Systems

(4 periods)

This unit on number systems is systematically introduced in the student book starting with decimal number system followed by binary, octal and finally hexadecimal base systems.

Information to teacher

You need to first introduce the students to decimal number system because this is the system they familiar with in mathematics. It is important to use practical examples to explain how the magnitude (weight) a number can be determined using absolute value, weight-by position (place value), and its base. It is important to help the learner understand that the base of octal and hexadecimal number system are in powers of two i.e. octal is 2^3 while hexadecimal is 2^4 . This explains why the two number systems are considered as shorthand for large binary numbers.

Materials: Scientific calculator, internet sites, charts illustrating number systems, and projector.

Preparation

Be prepared to handle weak students who have poor background in mathematics or physics. You may need to give such

student to more specialized attention using practical exercises.

Teaching guidelines 5.2

- Capture the attention of the class by making them carry our learning activities in the student's book. You may demonstrate how to convert number from one system to another using videos.
- Through class exercises, make sure each students participants in converting decimal numbers to binary, octal and hexadecimal forms.
- After introducing the students to the class activities, direct them to the review exercise in then student book to try out acquired knowledge and skills.

Converting Decimal to other Base systems

(2 periods)

Information to the teacher

This unit equips the learner with knowledge, skills and attitude required to convert decimal numbers to binary, octal and hexadecimal forms.

Having introduced the learners to the four number systems, let them try to convert decimal numbers to other base systems. It is from their response that you demonstrate how to perform decimal to binary, octal and hexadecimal conversions.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of calculations required. Let the students prepare well to make sure that they perform conversions correctly. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.3

- Through demonstration, take the student through examples and activities and in the student book to help them understand how to convert a binary number into other number systems
- Let the learners work on activities 5.4 to 5.7 from page 91 to 93 of the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Converting Binary to other Base systems

(2 periods)

This unit equips the learner with knowledge, skills and attitude required to convert binary numbers to their decimal, octal and hexadecimal equivalent.

Information to the teacher

Having introduced the learner to the four number systems, through discover or research let them try to convert binary numbers to other base systems. It is from their response that you demonstrate how to perform binary to decimal, octal and hexadecimal conversions.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct binary manipulations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.4

- Through demonstration, take the student through activities in the student book to help them understand how to convert a binary number into other number systems.
- Let the learners work on activities 5.8 and 5.9 on pages 94 and 96 respectively.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Octal to Decimal Conversion

(1 period)

This unit equips the learner with knowledge, skills and attitude required to convert octal numbers to decimal form.

Information to the teacher

Through discover or research let the learner try to convert octal numbers to decimal base system. It is from their response that you demonstrate how to convert octal to decimal numbers.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.5

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Individually or in groups, let the learners work on exercises 5.5 on page 97 of the student's book.

- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Octal to Hexadecimal Conversion

(1 period)

This unit equips the learner with knowledge, skills and attitude required to convert octal numbers to their hexadecimal.

Information to the teacher

It is important to note that conversion from octal to hexadecimal is a can easily be performed indirectly. Demonstrate to the learners how to convert an octal number by first converting it to decimal or binary form. After the demo, let the learner try to convert octal numbers to hexadecimal base system. .

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to hexadecimal computations.

Teaching guidelines 5.6

- Through demonstration, take the student through examples and

activities in the student book to help them understand how to convert a binary number into other number systems

- Provide interactive assessment exercise and let the learners work on octal to hexadecimal conversion.
- If you are not satisfied with the student's answers, organize for remedial sessions to help the weak students. It is important that you evaluate competence of each learner before proceeding to the next unit.

Hexadecimal to decimal Conversion

(1 period)

This unit equips the learner with knowledge, skills and attitude required to convert hexadecimal numbers to their decimal equivalent.

Information to the teacher

Since the students have developed skills in converting octal number to decimal, let the students use the same approach to convert hexadecimal numbers to decimal form.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct hexadecimal to decimal computations.

Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.7

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Individually or in groups, let the learners work on assessment exercise 5.6 on page 99 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Decimal fraction to Binary Conversion

(1 period)

This section gives learners detailed account on how to convert floating point (fractional) decimal number to binary form.

Information to the teacher

Before you introduce this unit, use mathematics background of representing a real number in standard form. It is from this knowledge that you introduce

Binary fraction to decimal Conversion

the concept of floating point as used in computing. This will help the learner appreciate how computers represent a real number using integer and floating point (fractional) parts.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they correctly convert decimal fraction to binary form. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.8

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Individually or in groups, let the learners work on activity 5.10 on page 101 of the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

(1 period)

It is important for the learner to convert binary fractions to decimal forms.

Information to the teacher

This section demonstrates to the learner how to convert floating point binary numbers to to decimal form.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they correctly convert decimal fraction to binary form. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.9

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Individually or in groups, let the learners work on activity 5.11 and assessment exercise 5.7 on page 102 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help

the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Negative Decimal to Binary Conversion

(1 period)

This section demonstrates how to convert negative decimal numbers to binary using ones and twos complement. It is important that the teacher introduces this section using simple examples.

Information to teacher

When introducing ones and twos complements let the students understand that the term complement is also applicable in other number systems. For example, you can demonstrate how to complement in decimal numbers. It is after this that you can demonstrate how to convert negative integers to binary using ones and twos complement.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they correctly convert fractional decimal numbers to binary form. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.10

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Individually or in groups, let the learners work on activities 5.12 and 5.13 (pages 102 and 103) in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Arithmetic Operations on Binary Numbers

(2 periods)

This section demonstrates to the learner how to perform arithmetic operations on binary numbers. It is important that the teacher introduces this topic using simple operations.

Information to teacher

Like in mathematics, this section demonstrates how to perform binary addition, subtraction, division and multiplication. But before you cover this topic, let the students first perform the four basic arithmetic operations, i.e. addition, subtraction, division and multiplication. In each case, use examples

to draw similarities between binary and decimal arithmetic operation.

Materials: Student book, calculator, web tutorial, and charts showing summary of binary to other number conversions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they correctly convert fractional decimal numbers to binary form. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 5.11

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems
- Let the learners work on activities 5.14 to 5.20 from page 104-109.
- Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for Unit 5

Assessment Exercise 5.1 (page 87)

I. Definition of terms

- Bit (Binary digit) refers to the two basic units digits 0 and 1 used to represent data
- A byte is a group of eight bits

used to represent a data

- A nibble is a 4-bit hexadecimal digit which is usually half a byte.
- A byte consists of 8 bytes while nibble is a grouping of 4 bits
 - Digital electronic devices that utilise binary system such as digital cameras, TVs and media players are dominating the electronic market.
 - Analog signal is represented using continuous sinusoid wave while digital signal is represented using square wave.

Assessment Exercise 5.2 (page 94)

- (a) 5_{10} (b) 15_{10} (c) 1390_{10}
(d) 191_{10} (e) 89_{10} (f) 455_{10}

Assessment Exercise 5.3 (page 95)

- (a) 244_8 (b) 247_8 (c) 162_8
(d) 565_8 (e) 222_8 (f) 3370_8
(g) 1413_8 (h) 1331_8 (i) 6347_8
(j) 46653_8

Assessment Exercise 5.4 (page 97)

- (a) 10100100 (b) 10100111
(c) 10100010 (d) 101110101
(e) 10010010 (f) 11011111000
(g) 1100001011 (h) 1011011001
(i) 110011100111
(j) 100110110101011

Assessment Exercise 5.5 (page 97)

- The student should use group of four bits for every hexadecimal digit

to get the following:

- (a) 1010010100
- (b) 1001000111
- (c) 11101111101
- (d) 100100000111
- (e) 101110101101
- (f) 110010111101111
- (g) 101100001011100
- (h) 1010101111001101
- (i) 1010101110000111010
- (j) 11001100010001000111001111

2. The student should group the number to four bits from right to left to get IE37.

Assessment Exercise 5.6 (page 99)

- (a) 63_{10} (b) 52_{10} (c) 66_{10}
- (d) 640_{10} (e) 512_{10} (f) 123_1
- (g) 83_{10} (h) 511_{10} (i) 229_{10}
- (j) 118_{10}

Assessment Exercise 5.7 (page 102)

- (a) 0.59375_{10} (b) 0.125_{10} (c) 0.65625_{10}
- (d) 3.375_{10} (e) 5.9375_{10} (f) 4.75_{10}

Assessment Exercise 5.8 (page 105)

- 1. 10001 2. 11101 3. 11001
- 4. 110011 5. 1011001 6. 10010
- 7. 1001010 8. 1010011
- 9. 10111110 10. 1000000
- 11. 1101101 12. 10100110

Assessment Exercise 5.9 (page 106)

- 1. 1111 2. 001 3. 10100

- 4. 1001_2 5. 110101 6. 11111
- 7. 10 8. 1010010
- 9. 11000110 10. 10001

Assessment Exercise 5.10 (page 110)

- 1. 11111001₂
- 2. (a) 11100001 (b) 11100100
(c) 1111011
- 3. (a) 111 (b) 10000_2 (c) 1
(d) 110010_2

Guideline to Activities

Activity 5.1: Magnitude of numbers (page 87)

The learner is expected to define 485 in terms of base value as a decimal number. Each digit has absolute value regardless of its position. The place value makes 4 on the left have more weight than 8 and 5 because it is under place value of hundreds.

Activity 5.2: Types of number systems (page 88)

The learner is expected to understand the four number systems already provided in the student book from page 87-89. These are binary, decimal, octal and hexadecimal systems.

Activity 5.3: Octal and hexadecimal number systems (page 89)

This activity requires the learner to research on the Internet reasons for use of octal and hexadecimal systems. Some

of the reasons include shortening long strings of binary numbers, simplifying programming, and processor and memory optimization.

Activity 5.4: Converting decimals to binary form (page 91)

This activity requires the learner to convert a decimal number.

247 to binary form i.e 1111011_2 .

Activity 5.5: Decimal to binary conversion (page 92)

1. The first activity requires the learner to use place value method to obtain the following binary numbers:

- (i) $145 = 10010001$
- (ii) $1280 = 10100000000$
- (iii) $5204 = 1010001010100$
- (iv) $8000 = 1111101000000$

2. The first activity requires the learner to use both division by base and place value method to obtain the following binary numbers:

- a) $10 = 1010$ b) $512 = 1000000000$
- c) $43 = 101011$ d) $143 = 10001111$
- e) $365 = 101101101$ f) $954 = 1110111010$

Activity 5.6: Decimal to octal conversion (page 93)

The first activity requires the learner to use division-by-base method to obtain the following octal numbers:

- (a) $999 = 143$ (b) $1875 = 3523$
- (c) $5210 = 12132$ (d) $505 = 771$
- (e) $1810 = 3422$ (f) $3185 = 6161$
- (g) $1000 = 1750$ (h) $750 = 1356$

Activity 5.7: Decimal to hexadecimal conversions (page 93)

The first activity requires the learner to use division-by-base method to obtain the following hexadecimal numbers:

- (a) $107 = 6B$ (b) $9850 = 267A$
- (c) $5207 = 1457$ (d) $7500 = 1D4C$
- (e) $7075 = 1BA3$

Activity 5.8: Binary to decimal conversions (page 94)

The first activity requires the learner to convert binary numbers to obtain the following decimal numbers:

- 1. $100100 = 36$
- 2. $1011110 = 94$
- 3. $11111111 = 255$

Activity 5.9: Binary to hexadecimal conversion (page 96)

The first activity requires the learner to convert binary numbers to obtain the following hexadecimal numbers:

- 1. $10111100110 = DE6$
- 2. $111011011 = 1DB$
- 3. $110111 = 37$
- 4. $0101110 = 2E$

Activity 5.10 Decimal fraction to binary conversion (page 101)

1. The first activity requires the learner to convert a real number with decimal points to obtain the following binary numbers:

$$43.5625 = 101011.1001$$

2. The first activity requires the learner to convert the following real number to obtain the following binary numbers:

$$(a) 0.625 = 0.101$$

$$(b) 0.450 = 0.01110011\dots$$

$$(c) 2.500 = 101.1$$

$$(d) 5.1625 = 101.001010011\dots$$

$$(e) 7.1875 = 111.0011$$

$$(f) 0.350 = 0.010110011\dots$$

Activity 5.11: Binary fraction to decimal conversion (page 102)

The first activity requires the learner to convert the following a binary number with floating point to obtain the following decimal number:

$$11.11011 = 3.84375$$

Activity 5.12: Ones complement (page 102)

The first activity requires the learner to convert the following a binary number to obtain the following ones complement:

$$(a) 1101001 = 0010110 \Rightarrow 22$$

$$(b) 1111010 = 0000101 \Rightarrow 5$$

$$(c) 10101101 = 01010010 \Rightarrow 82$$

$$(d) 1011111 = 0100000 \Rightarrow 32$$

$$(e) 1011001 = 0100110 \Rightarrow 38$$

$$(f) 11100111 = 00011000 \Rightarrow 24$$

Activity 5.13: One's and two's complement (page 103)

1. The first activity requires the learner to demonstrate how to represent decimal number like 945 using nine's complement. The second bullet is conversion of 11010010 two's complement gives $00101101+1 = 101110$

2. The first activity requires the learner to convert negative decimal numbers to the following binary equivalence using at least 8-bit twos complement:

$$(a) -20 \Rightarrow 11101100$$

$$(b) -55 = 11001001$$

$$(c) -108 = 10010100$$

$$(d) -586 = 0110110110$$

Activity 5.14: Binary addition (page 104)

This activity requires the learner to understand the procedure of adding two binary numbers to obtain

$$100011.$$

Activity 5.15: Binary addition (page 104)

This activity requires the learner to understand the procedure of adding three binary numbers to obtain 101000.

Activity 5.16: Binary subtraction (page 106)

This activity requires the learner to understand the procedure for direct subtraction of binary numbers to obtain the following:

- (a) $10011 - 1100 = 111$
- (b) $10110 - 1011 = 1011$
- (c) $101 - 100 = 1$
- (d) $10111 - 1111 = 1000$

Activity 5.17: Subtraction using one's compliments (page 107)

This activity requires the learner to understand the procedure for converting -13 to binary form, i.e. 1110010.

Activity 5.15: Subtraction using two's complement (page 107)

This activity requires the learner to convert the decimal numbers to binary form before calculating the differences using twos complement to obtain:

- (a) $11111 - 10001 = 1110$
- (b) $11011 - 101 = 10110$
- (c) $1111111 - 110010 = 1001101$
- (d) $10001 - 100011 = 11101110$

Activity 5.19: Binary Multiplication (page 108)

This activity requires the learner to understand the procedure for direct

multiplication of binary numbers to obtain the following:

- (a) $101101 \times 110 = 100001110$
- (b) $101101 \times 111 = 100111011$
- (c) $1011.01 \times 110.1 = 1001001.001$

Activity 5.20: Binary division (page 109)

This activity requires the learner to understand the procedure for direct division of binary numbers to obtain the following:

- (a) $1011 \div 11 = 11 \text{ rem } 10$
- (b) $10011 \div 101 = 11 \text{ rem } 100$
- (c) $1111 \div 11 = 101$
- (d) $11 \div 11 = 1$

Unit Test 5 (page 110)

1. (a) *Octal number system* is a base 8 system while the *decimal number system* is a base 10 system.
(b) *Binary number system* is a base 2 system while *hexadecimal number system* is base 16 system.
2. (a) 46_{10} (b) 43_{10} (c) 6_{10}
3. (a) 1100010101 (b) 1000111010
(c) 101010 (d) 11110
4. (a) 10101 (b) 101 (c) 11100
(d) 1111 (e) 1010
5. (a) 1100_2 (b) 011_2
(c) 11100_2 (d) 1101000_2
(e) 10001000001_2

6.

	Decimal	One's	Two's
(a)	-75	0110100	0110101
(b)	-80	0101111	0110000
(c)	-100	0011011	0011100

7. (a) 11100_2 (b) 111001_2

8. (a) 0.5625_{10} (b) 5.75_{10}
 (c) 3.625_{10} (d) 0.125_{10}

9. (a) 0.010110 (b) 10.1
 (c) 1000001.0011_2 (d) 10001.001_2

Computer System and Maintenance

Computer Arithmetic

Unit 6

Boolean Algebra and Logic Gates

Student's Book page 101 – 131. (18 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

1. Identify different logic gates, theorems of Boolean algebra and evaluate Boolean expressions.
2. Utilize laws of Boolean algebra on Boolean expressions and draw a simple logic circuit using logic gates.

Learning objectives

Knowledge and understanding

- Identify logic operators, Truth tables and evaluate Boolean expression using Boolean laws.
- Describe the use of logic gates and apply Boolean laws on logic gates.

Skills

- Apply laws of Boolean algebra on Boolean expressions.
- Draw graphical representation of different logic gates and construct and evaluate a logic circuit.

Attitudes and values

- Appreciate the logical reasoning while using Boolean operators and laws applied on Boolean numbers

- Appreciate logic gates and laws applied on logic circuit

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence supports interpersonal respect and promotes national unity.

Links to other subjects

Mathematics (Logic)

Assessment criteria

Students utilize laws of Boolean algebra on Boolean expressions, draw logic circuit from given expression, derive correct expression from given circuit.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion
- Role play

Background information

This is one of the most enjoyable and easiest units in computer science yet students find it very difficult if proper introduction and step by step instruction is not given. Introduce this section with the simple concept of a light switch and that of a 1 and a 0, ON and OFF.

Suggested teaching/ learning activities

Circuits and Logic gates

(2 period)

Information to the teacher

Come up with easy IF conditions that lead to a certain action being taken or not. For example:

1. If it rains then pick your umbrella or
2. If you want to buy land, you must have enough money and a willing seller.

In so doing the student will quickly learn to associate conditions to outcomes,

just the way a logic gate switches ON or OFF based on the inputs.

Materials: whiteboard, projector, student text book.

Preparation

1. Carry with you pictures of the logic gates on a wall chart to class.
2. If you can get a bread board for constructing simple LED circuits the it would be better when demonstrating.

Additional information teacher

- Although this topic is difficult, try to simplify it as much as possible.

Teaching guidelines 6.1

- Start by asking one student to switch the light ON and OFF at the switch. Impress on the students the simplicity of the two state process - - the ON can be interpreted as a 1 while the OFF as a 0.
- Impress on the students the importance of this simple method of instruction representation.
- Now request the student to do learning activity 6.1 on page 111.
- Now cover the content in this section. For each circuit represented by Figure 6.1 - 6.4 on page 112 let the student clearly understand the logic of using switches to control how the bulb comes on and goes off. Remember this will enable them understand logic gates in the future periods.

Logic gates

(3 periods)

Information to teacher

Materials: student text book, wall charts, whiteboard, projector, movie, computer connected to internet, circuit bread board.

Preparation

Download a movie which shows how logic gates work. Alternatively identify a good website which has the relevant information that the students can access and do research.

Teaching guidelines 6.2

- Request the students to study Table 6.1 on page 114 as an activity. Let them do research on the internet too concerning each and every basic logic gate.
- Step by step with the learner, give real life examples using logic statement that reinforce the logic for each gate e.g. The following statements may be suitable for the gates suggested below:
 1. NOT: Ask the student to say the opposite of certain actions e.g. Stand ---> sit; Shout ----Whisper; Laugh ---> Cry, Start ---> Finish etc.
 2. OR: To qualify you need to have an identity card or a passport;
 3. AND: In order to withdraw money from the bank you need your identity card and to sign with the right signature

Cover the content in this section.

Truth tables

(2 periods)

Information to teacher

Materials: student text book, wall charts, whiteboard, projector, movie, computer connected to internet.

Preparation

Download a movie which shows how logic gates work. Alternatively identify a good website which has the relevant information that the students can access and do research.

Teaching guidelines 6.2

- Request the students to compare the statements concerning each logic gate in Table 6.1 to its relevant truth table in Figure 6.5.
- After that, create simple two gate circuits and develop truth tables for each together with the students in class. This will make each student to understand the interaction between different gates.
- Guide the students to do learning activity 6.2 on page 115. This will not only help them to reason and apply their knowledge about the gates but will expose them to real world application of the logic gates i.e. let them know that the ICs they see on the motherboard for example have internal structures like the ones

depicted in Figure 6.6.

- Prompt the student to do learning activity 6.3 on page 116. Go round the class checking that each student is able to understand how the truth Table 6.2 was finally derived. This one on one session with the student will create an inclusive learning environment where you can help each student individually.
- Conclude the period by giving the students homework to do learning activity 6.4 on their own on page 117.

Solving problems using logic gates

(2 periods)

By the end of this section, the learner should be able to solve simple problems using logic gates.

Materials: A computer connected to the internet, projector, .

Preparation

Nearly every problem that requires control of a process which has inputs and outputs can be solved by designing relevant digital circuits. The aim of this section is to introduce the student to problem solving, creativity and automation when reasoning about how digital circuits can be used to control of solve real world problems.

Teaching guidelines 6.3

- Again start this period with the simplest control in a circuit - - - a

light switch. Draw a parallel between the analogy of a light switch and a NOT gate i.e. When the input is a 1 the output is a 0, just like when you close the switch the light comes ON and when you open it the light goes OFF. This two seem to be behaving opposite to each other i.e. I expect if the input is one, I should get the output as a 1 so that the light can come ON.

- Guide the student to do learning activity 6.5 on page 118 then they will understand that it needs two NOT gates in series with one another to create a digital light switch.
- Guide the students to do learning activity 6.6 on page 118. Go round the groups to see the progress of each group and give relevant guidance where need be.
- Come up with more scenarios for the student to do more practice.

Boolean algebra

(2 periods)

By the end of this section, the learner should be able to solve simplify and work out truth tables for Boolean algebra expressions.

Materials: a computer connected to the internet, student text book, whiteboard, projector.

Teaching guidelines 6.4

- Start by showing a link between boolean algebra and digital logic gates i.e. for both we reason in terms of 0s and 1s and we can construct truth tables.

- Go through the laws of boolean algebra and let each student listen and make relevant notes.
- Guide the student to do learning activities 6.7 and 6.8 on page 123 to 125. This will enable the students to develop confidence in boolean algebra.
- Teach the students one golden rule: two boolean algebra expressions are equivalent if their truth tables give the same output.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Sum of Product (SOP) and Product of Sum (POS)

SOP and POS are two sides of the same point ; Repeat the concept of the AND operation and how it can be seen as the product of binary values while the OR operation can be viewed as the binary SUM.

Sum of product

- Remind the student that in binary a 1 has a complement of a 0 and vice versa. Use the A and \bar{A} notation to represent a number and its complement.
- Remind the student about the two input AND. Now guide the student to understand that even when the inputs are more than two, the same laws hold i.e. all the inputs have to be 1 in order to get a 1 output.

- Guide the student to do Activity 6.9 on page 127.

Product of Sums

Guide the student to understand what POS is all about. Using Figure 6.14 as a reference point, guide the learner to verify all possible inputs and outputs of this arrangement.

- Guide the learner to do Activity 6.10 on page 128.
- Take the students through Activity 6.11 on page 129 so as to solidify the concepts learned.
- After this, give them Activity 6.12 on page 130 to gauge the level of their understanding. Appraise their progress in class and then guide the entire class through the steps of solving the challenge.

NAND and NORAS Universal Gates

Guide the student through activity 6.12 on page 130. Using relevant examples, justify why NAND and NOR gates are universal gates i.e. using this gates we can create a combination that can solve any existing need or problem. Use the examples previously covered in the book to demonstrate this concept too.

Answers

Assessment exercise 6.1 (page 120)

1. A logic gate is an electronic circuit that has some idealized or physical device implementing a Boolean function; that is, it performs a logical

operation on one or more logical inputs, and produces a single logical output

2. A truth table is a mathematical table used in boolean algebra or propositional logic to compute the outcome of all possible combinations of input values i.e. it can be used to tell whether an expression is valid for all legitimate input values.

3. 1
4. A Q
0 1
1 0

5. 1
6. A B Q
0 0 0
0 1 1
1 0 1
1 1 1

7. A NOR gate is an OR gate that has a NOT gate connected on its output therefore it inverts the OR gate output.

8. NAND gate is an AND gate that has a NOT gate connected to its output hence it inverts the output of the AND.

9. (i) We can decide to feed each gate with its own set of inputs separate from the other i.e. AB then CD as in when we input 00 at AB we do the same at CD. In so doing the output would be as follows:

A	B	C	D	E	F	Q
0	0	0	0	0	0	1
0	1	0	1	1	0	0

1 0 1 0 1 0 0
1 1 1 1 1 1 0

Alternatively the inputs can be sixteen as follows if we decide that all the four input ABCD will follow a four bit input order:

A	B	C	D	E	F	Q
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	1	1	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	1	0	0
1	0	0	1	1	1	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	0

9. (ii) A B C D E Q
0 0 0 1 0 1
0 0 1 1 0 1
0 1 0 1 0 1
0 1 1 1 1 0
1 0 0 0 0 1
1 0 1 0 0 1
1 1 0 0 0 1
1 1 1 0 1 1

12. Based on the facts provided we can deduce that the logic circuit has three inputs A, B and C and a single output Q. From what we have been

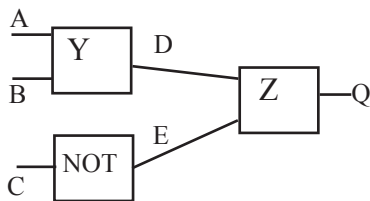
told:

A B C Q

0 0 0 1 (alarm raised)

1 1 0 1(alarm raised)

If the circuit has three inputs we immediately deduce that one is a NOT gate. Also the output Q is from a gate that has two inputs. We therefore draw as follows:



Every time $C=0$; then $E=1$ and $Q=1$ Regardless of whether $A=B=1$ Or $A=B=0$. We deduce as follows:

- (i) If Y is an AND gate then Z is and OR gate to satisfy the conditions given in the question.
- (ii) If Y is an OR gate then Z can only be an OR gate too in order to satisfy the conditions given in the question.

Guidelines to Activities

Activity 6.1: Switching a touch ON/OFF (pg. 111)

This activity enables the learner to conceptualise the two state nature of digital logic. Apart from a light torch you can also use a light switch in the room to demonstrate the concept. Make sure that the learner appreciates the following:

- i. The switch has two states Open / Closed.

- ii. When Open the voltage in the circuit is low hence bulb is Off.
- iii. When closed, the voltage in the circuit is high hence the bulb is On.
- iv. Other possible two state representations in the universe e.g. On/Off; High/Low; Hot/Cold and finally 1/0.

Activity 6.2: ICs and their internal logic gate structure (pg. 115)

This activity enables the learner to conceptualise how the simple logic gates are integrated together to form integrated circuits. This activity will open the eyes of the learner to the fact that all the electronic components that have ICs are actually built using individual logic gates one after another. Make sure that the learner appreciates the following:

- i. That each IC has a special number which can be referenced on a website or IC design book to find out how the gates are arranged within it. Let them touch and feel the ICs you have brought to class.
- ii. Where possible, carry a bread board (for connecting ICs) to class, a bulb and some dry cells in a holder and connecting wires so that you can demonstrate using the ICs how the bulb comes on when you have the proper connections on the bread board. This will help the students to further understand table 6.1 on page 114 and the truth tables of Figure 6.5 on page 115.

- iii. That given Figure 6.6 on page 115, you can know the output of each gate in the IC by biasing the inputs appropriately.

Activity 6.3: Examples of coming up with truth tables (pg. 116)

Make sure to guide the student step by step through this activity after initially giving them a challenge to read and understand it. Before guiding them through, pose a question asking whether any of the students is willing to explain to the others what they have read and understood.

Activity 6.4: Examples of logic gate identification from given truth table (pg. 117)

Follow the same steps as in 6.3 above although this time let the learners form pairs and go through the provided example first.

Activity 6.5: Examples of using a logic gate to construct a light switch (pg. 118)

This as an application activity meant to help the student develop problem solving skills. Allow the student to go through this activity individually or in pairs as you wish.

Activity 6.6: Solving real life problems (pg. 118-120)

This builds on 6.5 to help the student apply the knowledge in logic gates to solve real life problems. Help the student to understand that however challenging a problem is, it can be solved through

step by step analysis as presented in the learning activity.

Activity 6.7–6.8: Boolean algebra example (pg. 123 - 125)

The best way of approaching this is to make sure that the student understands the notations e.g. if $A = 1$ then $\bar{A} = 0$; that $(.)$ stands for AND operation and $(+)$ stands for OR operation etc.

Activity 6.9: Verifying the logic circuit in Figure 6.13 (pg. 127)

This activity moves the student one step higher in their understanding of logic gate combinations, truth tables and their related Boolean algebra.

Activity 6.10: Verifying the logic circuit in Figure 6.14 (pg. 128)

This activity moves the student one step higher in their understanding of logic gate combinations, truth tables and their related Boolean algebra.

Activity 6.11: Verifying the logic circuit in Figure 6.14 (pg. 129)

This activity moves the student one step higher in their understanding of logic gate combinations, truth tables and their related Boolean algebra in terms of SOP and POS.

Activity 6.12: POS logic circuit (pg. 130)

This activity moves the student one step higher in their understanding of logic gate combinations, truth tables and their related Boolean algebra in terms of POS.

Activity 6.13: NAND and NOR gates (pg. 130)

This activity moves the student one step higher in their understanding of logic gate combinations, truth tables and their related Boolean algebra in terms of NAND and NOR gate combinations as universal gates that can be used to design any logic circuit needed to solve any solvable problem.

Unit Test 6 (page 131)

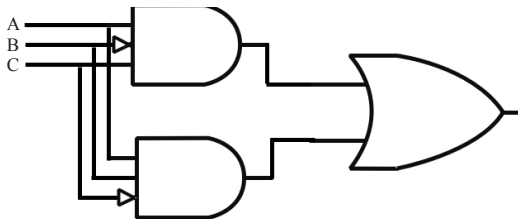
1. Yes. According to the identity law, $X+X = X$. Hence both statements can be written as: $X + YZ$.

2. NAND and NOR gates. using them, any possible logic circuit can be constructed.

3. SOP: all ANDed variables are ORed to get the required outputs; **POS:** all ORed variables are ANDed to get the final output.

4. (a)SOP circuit. It will consist of two AND gates each having an output feeding into an OR gate.

(b)



5. False. A minterm is an AND operation of variables. The one given is a maxterm.

6. False. A maxterm is an OR of variables. This is a minterm.

7. (i) Already simplified.

X	Y	Z		X.Y	Y.Z	F
0	0	0		0	0	
0	0	1		0	0	
0	1	0		0	0	
0	1	1		0	1	1
1	0	0		0	0	
1	0	1		0	0	
1	1	0		1	0	1
1	1	1		1	1	1

(ii) $F(X,Y) = (X+Y) + Y \cdot (X+Y)$

$F = (X+Y) + Y$ (using redundance law)

$F = X + Y + Y$

$F = X + Y$ (using identity law)

X	Y	X+Y	Y•(X+Y)		F
0	0	0	0		0
0	1	1	1		1
1	0	1	0		1
1	1	1	1		1

Notice $X+Y$ is equal to F in the truth table

Data Structure and Algorithm

Algorithms

Unit 7

Introduction to Computer Algorithm

Student's Book page 132–147 (24 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

- Identify appropriate steps to solve a problem.
- Identify an appropriate algorithm from a given problem.
- Represent graphically algorithm using flowchart.

Learning objectives

Knowledge and understanding

- Identify and explain the role, characteristics and structure of computer algorithm.
- Identify flowchart symbols and their meaning.
- Distinguish between variable and constant.
- Differentiate data types used in computer algorithm and identify memory size for each data type.
- Identify how data is stored in computer memory.
- Manipulate expressions in algorithm writing.
- Use reading and writing functions in algorithm.

Skills

- Trace an algorithm and predict output from a given input.
- Represent graphically the logic for a computer problem using flowchart.
- Able to use variable, constant and reading writing functions in computer algorithm.
- Able to evaluate an expression.

Attitudes and values

- Show concern in understanding steps to resolve computer problem using algorithm

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in four number systems. This include manipulating binary, decimal, octal and hexadecimal

numbers using arithmetic operators

Critical thinking: in most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires group work or pair work as covered in the student book unit 2, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of modern digital devices using binary logic. The learner is expected to transfer these skills to Boolean logic and digital electronics.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Critical thinking: Throughout this topic, we have emphasized on critical thinking in learning activities and assessment exercises.

Links to other subjects

This unit is highly linked to mathematics. This is because the learner is expected to apply mathematics skills in learned arithmetic, and algebra.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In activity 7.1 on page 135, we have addressed the issue of equitable access to education and food.
3. **Financial Education:** This is implied in various activities such as Activity 7.1 on page 135, 7.2 on page 136, 7.3 on page 138 and 7.10 on page 146 in which the student is expected to track loans and investments in the banking sector.

Assessment criteria

In this unit, the students are required to correctly derive the solution logic for a computational problem, represent it using flowchart, write algorithm, utilize variables and expressions in an algorithm. Beside activities and exercises provided in the student book, the teacher should use various assessment methods and tools to test whether the learner has acquired necessary knowledge, skills and attitude in problem solving.

Suggested teaching methodology

- Guided discovery
- Research

- Question and answer
- Discussion
- Role play

Background information

The most crucial stage in programming is algorithm design because it is at this stage that cost on maintenance can be minimised. Therefore, this unit is meant to introduce the students to basic principles of problem solving and design of algorithm. To test algorithms illustrated in student's book, we recommend use of trace tables and programming languages such as Pascal or C++. This is why we have deliberately avoided use of any programming language to implement the worked algorithms to leave it open for teachers and student to make their choice.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, question and answer method, group discussions and practical programming to nature problem solving skills, creativity and innovativeness.

Algorithm Concept

This section introduces the learner to basic concepts associated with algorithms. These include origin of the term algorithm, its definition, role and characteristics. Later we demonstrate how to express algorithms using natural language, pseudocode and flowchart.

Information to the teacher

This chapter is meant to introduce the

learner to basic principles of problem solving and design of algorithm. The teacher should use real life and practical examples such as recipe to explain how a computer follows sets of instructions to solve a computational problem. Make the learner appreciate that design, implementation and deployment of computer programs require creativity and problem solving skills acquired in related subjects like mathematics.

Materials: Computer, projector, website tutorial sites and other reference materials. You can also use wall charts depicting concept of a computer algorithm using real life example like recipe.

Preparation

Be prepared to handle weak students who have poor background in mathematics because algorithm design requires creativity, critical thinking and problem solving skills. You may need to give such student to more specialized attention using practical demonstrations and exercises.

Teaching guidelines 7.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out Activity 7.1 on page 135 in the Student's book. Let the group leader give each member an opportunity to give a

description as the secretary notes down the key points.

- The groups should present their algorithm in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners understand that algorithm is raw program that represents logical steps of solving a real life problem.

Additional information teacher

Having introduced the learner to fundamental concepts, encourage them to research principles of good algorithm design. Let the students also compare notes on how such algorithms may be converted into a real computer program using programming languages.

Design of Algorithms

(2 periods)

This unit equips the learner with knowledge, skills and attitude required to convert binary numbers to their decimal, octal and hexadecimal equivalent.

Information to the teacher

In this topic, we have briefly discussed common tools used to formulate an algorithm. Using simple examples, the teacher should demonstrate how to design an algorithm using natural languages, pseudocode and flowcharts. It is important to emphasize on standard symbols, keywords and convention used in each of the design tools. As a class or group activity, let the students draw

comparisons between flowchart and pseudocode. We encourage the teacher to use brainstorming sessions to guide students in identify advantages and disadvantages of other each algorithm design tool discussed in the student's book.

Materials: Computer, projector, chart showing sample algorithm, tutorial sites and other reference materials.

Preparation

Prior to introducing the topic, ensure that you have dry run your algorithm using trace table to avoid logic errors. You may also you create a sample programs to demonstrate output of the algorithm designs.

Teaching guidelines 7.3

- Through demonstration, take the student through examples and activities and in the student book to help them understand how to convert a real life problem into an algorithm.
- Individually or in groups, let the learners work on exercises 7.2 on page 136, and 7.3 on page 138 of the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Variables

(1 period)

This unit introduces the learner to variables as memory management concept in programming context. This requires students to have prior knowledge of computer architecture introduced earlier in unit two.

Information to the teacher

In the syllabus, this unit comes earlier than introduction to programming but we strongly feel that the teacher should introduce the students to programming languages such as Pascal, C and C++. This will help them be able to identify variables, constants, data types. This approach also simplifies delivery of this topic that requires students to identify variables, data types and memory size required by various data types.

Materials: Computer, projector, chart showing how computer memory is divided into cells, tutorial sites and other reference materials.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 7.4

- Through demonstration, take the student through examples and

activities in the student book to help them understand how to use variables in algorithms.

- Individually or in groups, let the learners work on activity 7.4 on page 141 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Constants

(1 period)

This unit introduces the learner to constants as used in programming context. This requires students to have prior knowledge of computer architecture introduced earlier in unit two.

Information to the teacher

Although this unit comes earlier than introduction to programming, we strongly feel that the teacher should introduce the students to basic concepts of programming languages such as Pascal, C and C++. This will help them be able to familiarize them with constants. This approach will also simplify delivery of this topic that requires students to use constants such as pi, tax rate and speed of light.

Materials: Computer, projector, chart showing how to represent constants, tutorial sites and other reference materials.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they represent constants in algorithms.

Teaching guidelines 7.5

- Through demonstration, take the student through examples and activities in the student book to help them understand how to use constants
- Individually or in groups, let the students work on activity 7.5 and 7.6 on pages 142 and 143 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Operators and Expressions

(1 period)

Although there is no reference to specific language, this topic introduces the learner to mathematical operators and expressions. These include assignment, arithmetic, relational and logical operators and expressions.

Information to the teacher

Given that this topic does not require prior knowledge on any programming language, the teacher may be required to make reference to how the operators apply in mathematics or using sample expressions in spreadsheet program. However, we find it difficult to entirely discuss the entire topic without demonstrating the same in a programming environment. For this reason, we encourage the teacher to use a language such as C++ to help the students understand how to create expressions using the four categories of operators and expressions.

Materials: Computer, projector, chart showing how computer memory is divided into cells, tutorial sites and other reference materials.

Preparation

This section has a lot of mathematical expressions. Let the students prepare well to make sure that they write correct expressions. Before the period, prepare sample spreadsheet show sample expression in each type operator.

Teaching guidelines 7.6

- Through demonstration, take the student through examples and activities 7.7 to 7.9 from page 143 to 144 in the student book. Let the students demonstrate understanding of operators and operator precedence using mathematical expressions
- Individually or in groups, let the

learners work on activities in the student's book.

- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Read and Write Functions

(2 periods)

This unit introduces the learners to concepts of read and write functions, used in pseudocode design,

Information to the teacher

In this topic, we dwell on the concept of top-down approach in which a large program is broken down into subprogram called functions, modules, subprograms or procedures. Give the student a practical activity to find out terms used to implement read and write functions.

Materials: Student book, calculator, web tutorial, and charts showing summary of read and write functions.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they correctly convert decimal fraction to binary form. Sensitize them on the need to have electronic calculators to simplify some

computations.

Teaching guidelines 7.7

- Through demonstration, take the student through examples and activities in the student book to help them understand how to use read and write functions.
- Individually or in groups, let the learners work on activity 7.10 on page 146 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for unit 7

Assessment Exercise 7.1 (page 138)

1. A flowchart is a graphical representation of an algorithm while a pseudocode is a set of structured statements. For advantages and disadvantages, the student are advised to explore beyond content provided in the student's book. She may use natural language, flowcharts or pseudocode.
2. Standard symbols used in flowchart design are rounded rectangle (start/stop), diamond (decision), rectangle (process), arrow (flow) and circle (off-line connector).
3. Decision symbol indicates that a condition has to be evaluated for an

alternative action to be performed while a connector indicates continuation of a flowchart to another section or page.

4. In simple programs that do not require conditional logic; if the program is large and modular; when communicating functionality of a system with non-programmers.
5. The following are advantages of using flowcharts over pseudocode:
 - Flowcharts are better way of communicating the system logic
 - With a flowchart, problem can be analysed in a more effective way.
 - Graphical representation of design serves as good program documentation
 - Flowchart makes it easier to debug and maintain a program
6. The registration process may be represented using natural language or pseudocode to get member details.

Assessment Exercise 7.2 (page 147)

1. Pseudocode for equation: $y = ax^2 + bx + c$
2. Algorithm for comparing x , y and z to return the least:

```
BEGIN
PRINT Enter Values for a,b and c
READ a, b, c;
PRINT Enter Value for x
READ x
 $y = ax^2 + bx + c$ 
PRINT (x, y) coordinate
END.
```

3. The read function is used to get user input from keyboard while

write is used to display output on the standard console like computer monitor.

4. Problem to calculate cost of depositing 200,000 after n years.

```
BEGIN
PRINT Enter deposit
READ deposit
PRINT Enter interest rate
READ rate;
Amount = deposit x (1.03)n
PRINT Amount
END.
```

Guideline to Activities

Activity 7.1: Natural Language Algorithm (page 135)

1. This activity requires the learner to outline the step-by-step procedure for waking up and preparing for class using natural language such as English or Kinyarwanda.
2. This activity requires the learner to outline the step-by-step procedure for preparing Ibiza or bugali using natural language such as English or Kinyarwanda.
3. This activity requires the learner to outline the step-by-step procedure for a program that may be used to calculate gross salary, net and deductions using natural language such as English or Kinyarwanda such as outlined below:

Prompt for basic salary, house allowance
commuter allowance and overtime

Calculate gross salary

Calculate 15% PAYE and 2.5% medical cover deductions

Calculate net salary by deducting the PAYE and medical cover amounts

Display the gross salary, net pay and deductions

Activity 7.2: Expressing Algorithm using pseudocode (page 136)

This activity requires the learner to create a pseudocode for calculating cumulative principal amount each year for five years using the formula:

$$A = P \times (1 + \text{rate}/100)^t$$

Activity 7.3: Expressing algorithm using flowcharts (page 138)

1. This activity requires the learner to clearly differentiate between the pseudocode as structured statements that resemble program code and flowchart that uses graphical symbols.
2. This activity requires the learner to draw a flowchart for calculating cumulative principal amount each year for five years using the formula in the process box:

$$A = P \times (1 + 8/100)^t$$

Activity 7.4: Declaring variables (page 141)

1. In the first expressions, the variables

are y, m and x while in the second expression, the variables are a, b and x where a and b are coefficients of x.

2. This activity requires the learner to identify input variables => L and w, and output variable, i.e. Area and perimeter.

Activity 7.5: Definition of constants (page 142)

The learner is expected to identify c as the constant in the first expressions.

Activity 7.6: Declaring constants (page 143)

This activity is a self-guided activity that requires the learner to use search engine to search for relevant content on constants in C++.

Activity 7.7: Operators and expressions (page 143)

This activity requires the learner to master the precedence rule in order to outline the correct order of evaluating the mathematical expression. In this case, the learners need to note that the evaluation starts with division and multiplication from followed by addition and subtraction

Activity 7.8: Logical operators (page 144)

This activity requires the learner to

appreciate importance of using logical operators to form complex conditional logic as follows:

```
IF (alarm-sounds AND time > 18:00  
AND day != holiday OR isweekend)  
THEN
```

Call the police.

Activity 7.9: Bitwise operators (page 144)

1. This activity requires the learner to differentiate between logic operator and bitwise operators in terms of values they operate. Logic operators work on decimal numbers while bitwise operators work on binary numbers bit-by-bit starting from left to right.
2. The answers to the bitwise operations on the table are: $1 \text{ and } 1 \Rightarrow 1$; $1 \text{ and } 0 \Rightarrow 0$; $0 \text{ and } 0 \Rightarrow 0$; $1 \text{ or } 1 \Rightarrow 1$; $1 \text{ or } 0 \Rightarrow 1$; $0 \text{ or } 0 \Rightarrow 0$.

Activity 7.10: Read and write functions (page 146)

1. This activity requires the learner to use read and write function in an algorithm that calculates roots of a quadratic equation. Below is a sample pseudocode:

```
BEGIN
```

```
WRITE "Enter values for a, b, x and c"
```

```
READ a, b, x, c
```

```
Root1 = (-b + SQRT(b2- 4 x a x c)) / (2 x a)
```

```
Root2 = (-b - SQRT(b2- 4 x a x c)) / (2 x a)
```

```
WRITE Root1, Root2
```

```
END
```

2. This activity requires the learner to write a read and write function in an algorithm that calculates loan repayment amount in four years' time.

```
BEGIN
```

```
SET rate = 0.12
```

```
WRITE "Enter loan amount and  
repayment period"
```

```
READ loan, years
```

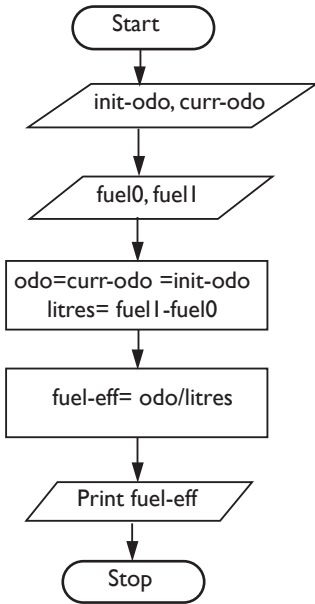
```
Amount = P x (1 + rate)t
```

```
WRITE amount
```

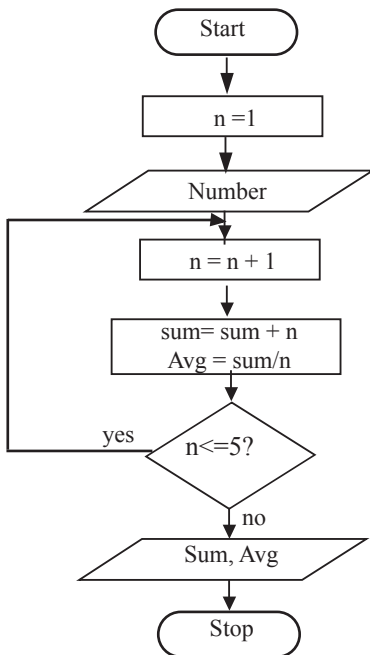
```
END
```

Unit Test 7 (page 147)

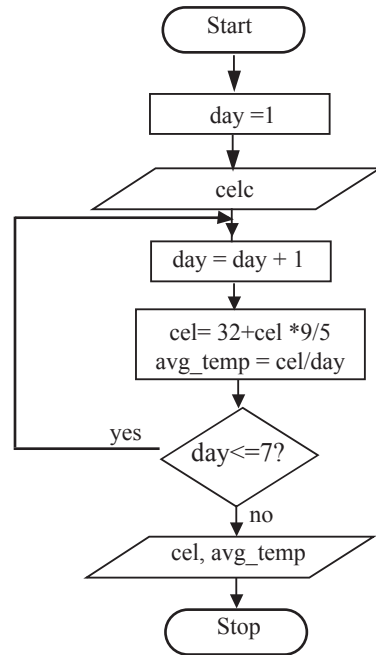
1. Definition of terms
 - a) precedence rule - established rule that assigns priority of each operator used in an expression
 - b) Variables - identifier that represents data values that changes
2. Flowchart for and displaying the estimated consumption in km/litre.



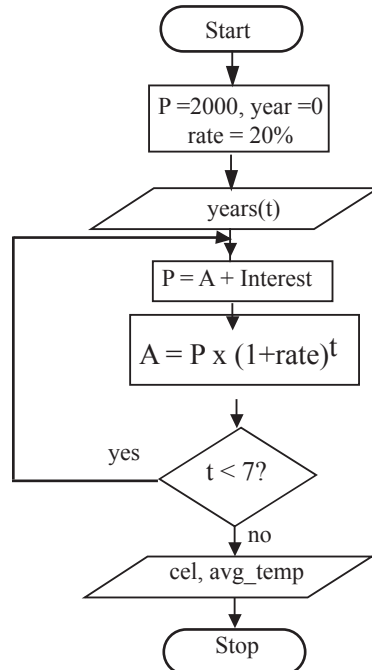
3. Flowchart for computing sum and average of five numbers: .



4. Flowchart for reading temperature in celsius and converting to fahrenheit:



5. Flowchart to calculate principal amount paid after 7 years



Data Structures and Algorithms

Algorithms

Unit 8

Control Structures and One Dimensions Arrays

Student's Book page 148 – 170. (12 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

- Derive logic in algorithm which include control statements.
- Handle one dimensional array in algorithm.

Learning objectives

Knowledge and understanding

- Identify and explain control statement in algorithm
- Explain the use of one dimension array data structure in algorithm.

Skills

- Able to use control statements in algorithm.
- Able to use one dimension array data structure in algorithm.

Attitudes and values

- Appreciate use of control statements in writing an algorithm.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in computer programming.

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Critical thinking: Throughout this topic, we have emphasized on critical thinking in learning activities and assessment exercises.

Links to other subjects

Most of the concepts in this unit are linked to mathematics. This is because the learner is expected to apply mathematics skills in designing control structures.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In activity 7.12, we have addressed the issue of equitable access to healthcare through electronic means.
3. **Financial Education:** This is indirectly implied in activity 8.6 that keeps determines discount rate of an item.

Assessment criteria

The students should be able to reasonably derive logic where

conditional statements, looping control structure and one dimensional arrays are required in solving a problem. Apart from activities and exercises provided in the student book, the teacher should use other assessment methods and tools to test whether the learner has acquired necessary knowledge, skills and attitude in solving problems that require use of control structures and one dimensional array.

Suggested teaching methodology

- Research
- Question and answer
- Discussion

Background information

This unit builds on computer algorithms to equip the learner with knowledge and skills on how to design algorithms using condition logic implemented using selection and looping control structures. To test the logic of each control structure illustrated in the student's book, we recommend use of trace tables and programming languages. Similar to approach used in the previous unit, we make no reference to specific programming language.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, question and answer method, group discussions and problem solving exercises that involve use of control structures and algorithms.

Conditional Logic

(2 periods)

This topic provides a more detailed discussion on conditional logic starting with simple conditional logic then progressing to complex.

Information to the teacher

The teacher should demonstrate how to incorporate conditional logic in design of flowchart and pseudocode. Let the learner research on internet or relevant resource material on how different programming languages implement conditional logic. As a class or group activity, let the students draw comparisons between simple and compound statements. We encourage the teacher to use brainstorming sessions to guide .

Materials: Computer, projector, chart showing sample conditional logic, tutorial sites and other reference materials.

Preparation

Prior to introducing the topic, ensure that you have dry run your algorithm using trace table to avoid logic errors. You may also you create a sample programs to demonstrate output of the algorithms.

Teaching guidelines 8.1

- Through demonstration, take the student through examples and activities and in the student book to help them understand how to represent a compound conditional

logic into other number systems

- Individually or in groups, let the learners work practice on formulating conditional logic from real life cases.
- It would be important to organize for remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Control Structures

(2 periods)

This section contains an overview of three control structures namely; sequence, selection and repetition supported by most programming languages. The section also serves as a footprint to sections that demonstrate use of conditional logic in selection and looping structures.

Information to the teacher

To widen the learner's understanding of various selection and iteration concepts, we suggest that the teacher provide more activities and exercises to complement what is been provided at the end of every unit. Where necessary, illustrations or practical demonstration should be used to help the learners understand difficult concepts. Your approach should make the learner appreciate that; design, implementation and deployment of most computer programs require use of this combination of the three control structures.

Materials: Computer, projector, digital materials. You may also use wall charts depicting concept of selection and iteration.

Preparation

Be prepared to handle weak students who have poor background in mathematics because algorithm design requires creativity, critical thinking and problem solving skills. You may need to give such student to more specialized attention using practical demonstrations and exercises.

Teaching guidelines 8.2

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out the activities related to the content during the period. Let the group leader give each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners understand that selection and control structures

are instrumental in day-to-day's problem solving.

Sequence control structure

(1 period)

Because sequence is the most natural way a computer executes a program, this topic gives a brief overview of sequence control structure using a pseudocode and a flowchart. Note that in the student's book, no more emphasis is given to sequence because the main focus in this topic is on selection and looping controls.

Information to teacher

Though this topic is not explicitly stated in the content section of the syllabus, it is important to expose the learner to the three control structures used in structured programming. However, the teacher should not give more on sequence control because the main focus is selection and looping controls.

Materials: Computer, projector, digital materials. You may also use wall charts depicting how a computer executes selection statements.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate implementation of sequence control structure.

Teaching guidelines 8.3

- Capture the attention of the class by making them carry learning Activities 8.1 on page 150.
- Through class exercises, make sure each student participant in converting a real life problem or routine to an algorithm.
- After introducing the students to the class activities, direct them to the previous unit in which all the examples and activities are implemented using sequence control structure.

Selection control structure

(2 periods)

This topic provides a more detailed discussion of selection control structure starting with IF conditional logic then progressing to switch selection.

Information to the teacher

The teacher should demonstrate how to incorporate selection control structure in design of flowchart, pseudocode and decision table. Let the learner research on internet or relevant resource material on how different programming languages implement selection control structures. As a class or group activity, let the students draw comparisons between nested selection and switch statements. We encourage the teacher to use brainstorming sessions to guide students in identifying situations that demand use of the four selection control structures

discussed.

Materials: Computer, projector, chart showing sample algorithm, tutorial sites and other reference materials.

Preparation

Prior to introducing the topic, ensure that you have dry run your algorithm using trace table to avoid logic errors. You may also you create a sample programs to demonstrate output of the algorithms.

Teaching guidelines 8.4

- Through demonstration, take the student through examples and activities and in the student book to help the learners demonstrate use of the four selection control structures in algorithm design.
- Individually or in groups, let the learners work on activities 8.2 to 8.4 from page 153 to 157 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Looping Control Structure

(4 periods)

This topic provides a more detailed discussion of looping control structure

starting with the FOR looping control structure.

The teacher should demonstrate how to incorporate looping control structure in design of flowchart, pseudocode and decision table. Let the learner research on internet or relevant resource material on how different programming languages implement loop constructs. As a class or group activity, let the students draw comparisons between while and do..while statements. We also encourage the teacher to use brainstorming sessions to guide students in identifying situations that demand use of the three looping control structures.

Materials: Computer, projector, chart showing how iteration is implemented in structured programming languages.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 8.5

- Through demonstration, take the student through examples and activities in the student book to help the learners demonstrate understanding and use of the four looping control structures in algorithm design.
- Individually or in groups, let the learners work on activities 8.5 to

8.9 (pages 159-165) in the student's book.

- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

One dimensional arrays

(3 periods)

This topic briefly introduces the learner to one of the simplest complex data structures known as one-dimensional array also known as linear list.

Information to the teacher

Because this is the first time the learner is being introduced to complex data structure, use simple real life examples to explain how arrays store data in the memory. Because we find it to difficult to entirely discuss arrays without practical demonstration, we encourage the teacher to use a language such as Pascal to help the students clearly understand array concepts such as subscript, elements declaration, and initialization. From our teaching experience, students find topic on arrays and other complex data types more challenging. This is why the teach should use practical examples such as chessboard to represent a two-dimensional array also known as table or matrix

Materials: Computer, projector, chessboard, and chart showing how arrays are stored in computer memory.

Preparation

This section has a lot of mathematical expressions. Let the students prepare well to make sure that they write correct expressions. Before the period, prepare sample spreadsheet show sample expression in each type operator.

Teaching guidelines 8.6

- Through demonstration, take the student through examples and activities in the student book to help them understand how to represent one-dimensional array in algorithm design.
- Individually or in groups, let the learners work on activities 8.10 and 8.11 on page 167 and 169 respectively.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers to Assessment Exercise

Assessment Exercise 8.1 (page 166)

1. Selection branches to an alternative action if while iteration repeats existing statement for a finite number of times

2. Importance of the following control structures
 - If .. is used to execute statements under true block if the Boolean return true
 - Nested if - used to test multiple options e.g. in assigning grades
 - Switch case - used when this an alternative to nested if used to test multiple conditions under case
3. Three types looping control structures are while, repeat..until and for loop.
4. Algorithm for comparing x, y and z:

```
BEGIN
PRINT Enter x, y and z
READ x, y, z;
IF x < y Then
  IF x < z = Then;
    least = x
  ELSE
    least = z
ELSE
  IF y < z Then
    least = y
  ELSE
    least = z
ENDIF
PRINT least
END.
```

5. Selection control structure are if, if..else, nested if and switch ..case selection
6. This requires nested if with the outer testing the gender an the inner nested if calculating tax payable based on gross monthly income.

Assessment Exercise 8.2 (page 169)

1. The following is an array of 25 buses:
 - Scores : array[0..24] of integer;
2. *Customer array*
 Customer : array[0..24] of integer
 Scores: Array [10] = { 64, 25, 69, 67, 80 85}

(b)

```

BEGIN
SET Scores=Array[10]of Integer
SET Initial_value=0, last_value = 9
FOR Index=Initial_value TO last_value DO
    READ Scores[Index];
    Index = Index + 1
LOOP
FOR Index=Initial TO limit DO
    READ Scores[Index];
    Index:= Index + 1
END FOR
END.
    
```

3. Algorithm to convert a number from one base system another. This require use of integer division
4. The array has only 5 but populated by 7 elements.

Guideline to Activities

Activity 8.1 Sequential control structure (page 150)

1. This activity requires the learner to draw a flowchart or write pseudocode that takes length and width, then calculates area and perimeter as shown by the following pseudocode:

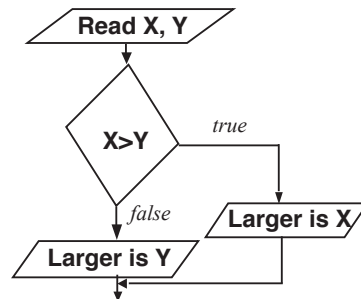
```

BEGIN
    PRINT Enter length of rectangle
    READ length;
    PRINT Enter width of rectangle
    READ width;
    area = length x width
    perimeter = 2x(length+width)
    PRINT area, perimeter
END.
    
```

2. The flowchart represents sequence control structure because the statements from input to output follows a sequential flow.

Activity 8.2: If ... else selection (page 153)

1. This activity requires the learner to draw a flowchart that takes two numbers as input and displays the larger of the two as shown in flowchart segment of Fig.8.1 below:



2. This activity requires the learner to use a trace table to determine the value of z as follows:
 - (a) If x= 20 and y=10, the decision returns true hence z=z+50 is executed hence output 150.
 - (b) If x= 19 and y=20, the decision returns false hence the output is 100.

Activity 8.3: Nested If selection

(page 155)

This activity requires the learner to study the conditional logic used to compare the three numbers and represent the logic in as demonstrated by assessment exercise 8.4 question 4.

Activity 8.4: Switch/case selection (page 157)

This activity requires the learner to draw a flowchart that takes two numbers as input and displays grades based on score obtained. The logic of this problem is similar to that demonstrated by Fig. 8.10 on page 156 of the student's book.

Activity 8.5: For loop (page 159)

This activity requires the learner to use for loop in an algorithm as shown in the following segment of a pseudocode:

```
FOR count = 0 TO 10 DO
```

```
    PRINT "Enter student mark"
```

```
    READ mark
```

```
    total = total+mark
```

```
END FOR
```

```
Average = total/count
```

Activity 8.6: While loop (page 160)

This activity test the learner's ability to use of the while looping control in a practical scenario. To control the electric alarm, the following pseudocode may be used to implement the beeping mechanism:

```
SET count = 0
```

```
WHILE count <20 DO
```

```
    BEEP alarm
```

```
    Count = count+1
```

```
END WHILE
```

Activity 8.7: Repeat until loop

(page 162)

This activity test the learner's ability to replace for loop with a repeat until logic construct shown in pseudocode segment below:

```
SET count = 0
```

```
REPEAT
```

```
    PRINT "Enter student mark"
```

```
    READ mark
```

```
    total = total + mark
```

```
    count = count + 1
```

```
UNTIL count >= 10
```

```
Average = total/count
```

Activity 8.8: Finite and infinite loop (page 163)

This activity test the learner's problem solving ability from the case study provided and be able to determine input, processing and output specifications before designing the algorithm.

Activity 8.9: Break and continue (page 165)

1. This activity tests the learner's ability to use the continue statements in algorithm that tests whether a number is prime. Refer the learners online resources that would guide

the learner on determining whether a number is prime or not.

2. This activity tests the learner's ability to use either break and continue statements in algorithm that tests whether a number is prime. Refer the learners to the student's book on page 163 for demonstration on how to use break and continue.

Activity 8.10: One dimensional array (page 167)

This activity test the learner's ability to represent and declare one-dimensional arrays:

1. a) Array name=>customer; data type =>integer; elements => 6
b) Array name=>temperature; data type =>real; elements => 6
2. Below is a sample declaration for each of the two arrays:
customer: Array[6] of integer
temperature: Array[6] of real

Activity 8.11: Array of integers (page 169)

3. This activity tests learner's problem solving ability in a practical scenario and represent responses as array of integer that may be declared as follows:

Unit Test 8 (page 170)

1. Switch case - used almost the same as nested of when this an alternative to nested if used to test multiple
2. Importance of while, for and repeat.. until:

- a) While: pretests a condition to repeatedly execute statements in the loop zero or more times
 - b) For: similar to while but mostly used on in case the number of repetitions is known before
 - b) Repeat...until: a post-test a loop that repeatedly executes statements in the loop at least.
3. The loop is infinite of condition for terminating the loop is never met.
 4. One dimensional is a list while a matrix is a two-dimension array.
 5. The following are factors to consider:
 - The name of the array - decide on suitable array name that indicates several elements are to be stored e.g. scores.
 - *Data type* of contents should be of the same type - floating point, integer, characters, strings.
 - Know the *size* of the array beforehand -the size of an array determines the maximum number of values that an array will hold.
 - This problem test whether a student has gained competence in using array variable.

PROGRAMMING

Procedural programming

Unit 9

Introduction to Computer Programming

Student's Book page 171 – 181. (6 Periods)

Key unit competence

By the end of this unit, the learner should be able to explain programming paradigms.

Learning objectives

Knowledge and understanding

- Classify different programming language generations.
- Outline different programming language paradigms.
- Point out features of good programming languages.

Skills

- Differentiate levels and characteristics of programming language.
- Classify programming languages according to their generations.
- Explain features of programming language paradigm.
- Outline characteristics of good programming language.

Attitudes and values

- Understand different perspectives of programming techniques

Generic competences addressed in this unit

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Critical thinking: Throughout this topic, we have emphasized on critical thinking in learning activities and assessment exercises.

Links to other subjects

Some of the concepts in this unit are linked to mathematics. This is because the learner is expected to apply knowledge and skills learned in number systems to understand how a computer program is translated to binary code.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In activity 9.1, we have brought in local East African languages such as Kinyarwanda and Kiswahili against foreign languages like Chinese, English and French.
3. **Financial Education:** This is indirectly implied in assessment exercise 9.1 on page 171 where we have attached value to local and foreign languages.

Assessment criteria

The students should be able to understand different computer languages and programming paradigms. Apart from activities and exercises provided in the student book, use other assessment techniques and tools to test whether the learner has acquired basic knowledge, skills basic in programming.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion
- Role play

Background information

Like mathematics, programming may be interesting to some students while others may find it challenging depending on the mode and timing of content delivery. According to our judgement on sequencing of content, we strongly feel that this topic seems to come late after introducing other technical concepts like control structures, arrays and algorithm. Furthermore, it is not possible to discuss about data types and their type without reference to programming languages. Therefore, we suggest that the topic be taught before introduction to high-level programming concepts covered in unit 7 and 8. This will help in providing practical demonstration where students find it difficult to grasp concepts such as variable declaration and initialization.

Suggested teaching/ learning activities

To introduce programming languages and paradigms, the teacher is expected to use demonstrations, question and answer method, group discussions and practical activities beyond those provided in the student's book.

Computer Programming Concepts

(1 period)

This gives an overview of basic programming concepts starting with the term computer program to compilers and interpreters. The section also serves as a footprint to sections that demonstrate use of create source code and translate it into an executable program logic.

Information to the teacher

There are fundamental concepts terms used in programming that should be clearly defined at this stage. For instance, understand the difference between a source code and object code; software and program; as well as compiler and interpreter. Using brainstorming and group works techniques let the students have a clear understanding of how a program is compiled from source code to get an executable file with .exe extension in Windows operating system. It is also important to use real life example to demonstrate how linking between the linker combine compiled source code with library functions.

Materials: Computer, projector, digital materials. You may also use wall charts depicting how a program is compiled from source code to object code.

Preparation

Be prepared to handle weak students who have poor background in mathematics because programming requires creativity, critical thinking and problem solving skills. You may need to give a weak learner to more specialized attention using demonstrations and practical exercises.

Teaching guidelines 9.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out Activity 9.1 and 9.2 on pages 171 and 173 respectively. Let the group leader accord each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners appreciate the role of programming solving in day-to-day's problems.

History of programming Languages

(1 period)

This topic gives an overview of how computer programming languages have evolved over since the time of first generation computers. Note that in the student's book, more emphasis is given to high-level programming languages that are easy to learn and use.

Information to teacher

To effectively deliver this topic, we suggest that you first let the students explore the historical perspective associated with electronic computers and how they were programmed. This will help the students appreciate how far technology has influence our modern life in which mobile phone are better off that the first generation computers.

Materials: Computer, projector, digital content, internet connectivity. You may also use wall charts to show how programming languages have evolved.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate implementation syntax of high-level programming languages.

Teaching guidelines 9.2

- Capture the attention of the class

by making them carry our learning Activity 9.3 on page 174 of the student's book.

- Through class exercises, make sure each students participants in converting a real life problem or routine to an algorithm.
- After introducing the students to the class activities, refer them to reliable online resources to explore on history of programming languages

High-level programming languages

(2 periods)

This topic provides a more detailed discussion of high-level languages classified into third, fourth and fifth generation languages.

Information to the teacher

Let the learner research on internet or relevant resource material on generations of computer language that marked departure from hardware oriented low-level languages. Where applicable, it is important to demonstrate and guide students practically on how to write a simple program like “hello world” using one of the languages in each category. This helps in exposing the learner to various programming environments which you may require to in the subsequent units.

Materials: Computer, projector, chart showing classifications of high-level languages.

Preparation

Prior to introducing the topic, ensure that you have sample program to easily distinguish high-level languages.

Teaching guidelines 9.3

- Through demonstration, take the student through examples and exercises let the students carry out research on how challenges of first and second generation languages influences evolution of high-level languages.
- Provide the learner with activities that will help them understand syntax of programming languages installed on computers in the lab.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Computer programming paradigms

(4 periods)

This topic provides serves as an introduction to four main programming paradigm.

Information to the teacher

The teacher should use practical examples to demonstrate differences in the four programming paradigm. Note that contrary to the syllabus

that classifies procedural as functional, we have corrected the anomaly by placing procedural paradigm as form of imperative programming. The teacher should clear the air on the differences between functions as used in languages such as C and C++ with function paradigm. Furthermore, the student should be made aware that more than one paradigm may be supported by a programming language. Let the student also distinguish between concepts like procedural against object oriented programming languages in terms of paradigm they support.

Materials: Computer, projector, digital content chart showing summary of programming paradigm.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 9.4

- Through demonstration, take the student through examples and activities in the student book to help them appreciate why the current paradigm shift is toward object-oriented programming.
- Individually or in groups, let the learners work on activities 9.4 on page 178 to 9.6 on page 180.
- If you are not satisfied with the

student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Features of good programming language

(1 period)

This topic highlights some of the features of a good programming language.

Information to teacher

To effectively deliver this topic, we suggest that you first let the students explore from reliable sources features of a good language. This will help the students appreciate factors that influence choice of a programming language in development of versatile Apps used on computers and mobile devices.

Materials: Computer, projector, internet. You may also use wall charts that summarize features of a good programming language.

Teaching guidelines 9.5

- Through class discussion, take the student through some of the desirable features of a programming language discussed in the student's book on page 179 and 180.
- Individually or in groups, let the learners work on activity 9.7 and exercise 9.4 on page 181.

Answers for Unit 9

Assessment Exercise 9.1 (page 173)

1. Definition of terms:
 - (a) Programming is the writing of computer programs
 - (b) *Source program* refers to a program that is not yet translated
 - (c) *object code* refers to the program code that is in machine readable form
2. A *compiler* translates the entire source program into machine readable form while an *interpreter* translates the source program to machine language line-by-line.
3. Software refers to program and its associated documentation while program is a set of instructions without documentation.

Assessment Exercise 9.2 (page 175)

1. Binary code are 0's and 1's, mnemonics are abbreviations and assemble converts a low-level assembly program to machine form.
2. First generation languages used machine code while second generation languages used mnemonics.
3. Plugboard having switches for "on" and "off" were used to electronically program a computer.
4. Advantages and disadvantages of low

level languages:

Advantages

- Program written using low level languages requires small amount of memory space.
- The processor executes them faster because they require minimal or no translation.
- Low level languages are stable and hardly crash or break down once written.

Disadvantages

- Low level languages are difficult and cumbersome to use and learn.
- They require highly trained experts both to develop and maintain.
- Checking for errors (debugging) low level programs is difficult and time consuming.
- Low level programs are machine dependent hence not portable.

5. Symbols on electric kettle:

- (a) Sands for bits 1 and 0
- (b) Computers only understand machine language hence every instruction must be converted to 0's and 1's.

Assessment Exercise 9.3(page 176)

1. Computer generations

- (a) Third generation make it possible to break down a program into components known as proce-

dures or modules

- (b) *Fourth generation* have advanced programming tools for integrating programs with databases, report generators and (GUI) applications

2. Evolution of programming languages:

- (a) First generation – machine languages.
- (b) Second generation – assembly languages.
- (c) Third generation – Pascal, C, COBOL, BASIC, FOTRAN, Ada.
- (d) Fourth generation – Visual Basic, Delphi Pascal, Visual COBOL etc.
- (e) Fifth generation languages – PROLOG, Mercury, LISP, etc.

3. Advantages and disadvantages of high-level languages:

Advantages

- High level languages are portable i.e. they are transferable from one computer to another.
- High level languages are user friendly and easy to use and learn.
- High level languages are more flexible, hence they enhance the creativity of the programmer and increase productivity in the workplace.
- A program in high level languages is easier to debug.

Disadvantages

- Their nature encourages use of many instructions in a word or statement hence the complexity

of these instructions cause slower program processing.

- They have to be interpreted or compiled to binary form before the computer can execute them.
 - They require large computer memory to run.
4. Students are expected to identify languages such as Pascal, *FORTRAN*, *COBOL*, *ALGOL*, C, and Ada.

Assessment Exercise 9.4 (page 181)

1. In OOP data and procedures are combined to form objects while logic program uses rules and facts to produce desired output.
2. Dictates choice of language to be used in terms of readability, write-ability and resource utilization.
3. *F#* supports functional programming paradigm.
4. To be able to advice on choosing the most appropriate language that can be used to implement an algorithm:

Guideline to Activities

Activity 9.1: Computer programming concepts (page 171)

This activity requires the learner to use background knowledge in linguistics to differentiate syntax from semantic. Syntax refers to set of rules that govern the structure of sentences in a given language while semantic is the meaning attached to words or sentence.

Activity 9.2: Computer programming (page 173)

This activity requires the learners to brainstorm in groups the relationship between mathematics and computer science, especially programming. The teacher should moderate the discussion in order to get the most appropriate responses from the groups. For example, some of the responses may be:

- Many of the functions and operators in all programming languages require some knowledge in mathematics.
- Computer sciences heavily rely on algorithms, which in turn heavily relies on mathematics.
- Formal methods is mathematically based technique for specification, verification and development of software

Let the learners appreciate the fact that a beginner in programming may not need mathematics, but as he/she advances through the level of difficulty, mathematics becomes important tool in problem solving.

Activity 9.3: Second generation programming languages (page 174)

This activity requires the learners to brainstorm in groups on assembly languages used on second generation computers for example *IBM7094* that was programmed using *Lisp*, *FORTRAN* and *COBOL* programming languages.

Activity 9.4: Programming paradigms (page 178)

This activity requires the learners to understand and explore further on programming paradigms discussed in the student book. One downloadable reference book that give thorough discussion of programming paradigms is “Concepts of Programming Languages, 10th edition”. The book is written by Robert Sebesta and published by Pearson Education, Inc.

Activity 9.5: Logic programming (page 179)

This activity requires the learners to understand and explore further on logic programming paradigms discussed in the student book from page 176 to 179. More on Logic programming can be available in chapter 16 of “Concepts of Programming Languages, 10th edition”.

Activity 9.6: OOP Paradigm (page 180)

1. This is an analytical problem that requires the learners to understand that OOP is not a distinct paradigm because it builds on imperative and functional programming. To avoid confusion or ambiguity, there need to be correction on the syllabus

to provide clear classification of paradigms especially OOP and procedure-oriented paradigms.

2. The second part of this activity requires the learner to master some basic concepts of OOP such as classes, inheritance and polymorphism (many forms). Let the learners distinguish between procedure-oriented and OOP programming paradigms.

Activity 9.7: Qualities of a good program (page 181)

This activity requires the learners to understand and explore further on qualities of good programming language discussed in the student book on page 179 and 180.

Unit Test 9 (page 181)

1. Program is set of instructions while software includes documentation.
2. Effort to improve simplicity, efficiency and resource utilization.
3. Smalltalk, Java, C++, Delphi pascal, Objective C, and C#.
4. In procedural programming, a program is broken down into procedures while in OOP, procedures and data are encapsulated into an object.

PROGRAMMING

Procedural Programming

Unit 10

Introduction to C++ Programming

Student's Book page 182 – 201. (12 Periods)

Key unit competence

By the end of this unit, the learner should be able to write and execute a given algorithm using C++ Programming language.

Learning objectives

Knowledge and understanding

- Describe the evolution of C++ language.
- Familiarize with C++ compiler environment.
- Identify steps followed to write a C++ program.
- Identify the use of input/output streams in C++ program.
- Differentiate different data types used in C++ program.
- Differentiate variable and constant in C++ program.
- Recall steps to execute a program.

Skills

- Apply the syntax of C++ language while writing a C++ program.
- Use cin and cout streams.
- Utilize variables and constants in C++ program.

- Write a sample C++ program and run it

Attitudes and values

- Derive algorithm for a given problem and implement the solution logic into C++ programming language.
- Read and interpret a simple C++ program containing input/output stream, variables and constants, and provide the intended results.

Generic competences addressed in this unit

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Links to other subjects

Most of the concepts in this unit are linked to mathematics. This is because the learner is expected to apply knowledge and skills learned in mathematics like calculating area of a rectangle to write C++ programs.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In activity 9.1, we have brought in local east African languages such as Kinyarwanda and Kiswahili against

foreign languages like Chinese, English and French.

3. **Financial Education:** This is indirectly implied in question number 12 of the Unit Test in which the student is expected to compute Malaika's loan repayments to a commercial banking.

Assessment criteria

The learner should be able convert a given algorithm into C++ program and execute the program. Apart from activities and exercises provided in the student book, use other assessment techniques and tools to assess whether the learner has acquired knowledge and skills C++ programming.

Suggested teaching methodology

- Research
- Question and answer
- Discussion

Background information

At this level, the learner is expected to be familiar with C++ programming languages introduced in the previous units. Having mastered basic concepts, the learner may be introduced to C++ environments using a simple example such as “Hello World” program. Though in our demonstration we have used DevC++ IDE, assist the learner in choosing their affordable C++ Integrated Development Environment (IDEs).

Suggested teaching/learning activities

To introduce C++ programming language, it is important to provide demonstrations and practical exercises to augment those provided in the student's book.

Evolution and Features of C++

(1 periods)

This topic gives an overview of how C++ evolved from "C with classes" to standard C++. The section also serves as footprint to sections that demonstrate how to write C++ programs.

Information to the teacher

To effectively deliver this topic, we suggest that students undertake activity 10.1 in the student book. It is from this background information that you can organize for brainstorming sessions to help students have a clear understanding of C++ syntax. It is also important to use demonstration differentiate the syntax of the two languages in terms of input, processing and output functions.

Materials: Computer, projector, C++ compiler, digital materials, internet connection. You may also use wall charts depicting how evolution of C++.

Preparation

Be prepared to handle weak students who have poor background in mathematics because programming requires creativity, critical thinking and problem solving skills. You may need to give a weak learner to more specialized attention using demonstrations and practical exercises.

Teaching guidelines 10.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out assessment exercise 10.1 on page 184 in the student's book. Let the group leader give each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners appreciate the role of C++ language as a system programming tool.

Syntax of C++ Program

(2 periods)

This topic introduces the learner to C++ syntax using globally accepted convention of "Hello world". Note that in the student's book, use of namespace instead of deprecated header file <iostream.h> or scope resolution operator.

Information to teacher

To effectively deliver this topic, we suggest that you first demonstrate

Compiling and Executing C++ Program

to the learner how to write, compile and execute hello world program. It is after this that you can give them more practical exercises for them to appreciate sensitivity of the compiler even on very minor syntax errors like punctuation or use of uppercase on keywords.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate implementation syntax of high-level programming languages.

Teaching guidelines 10.2

- Capture the attention of the class by making them carry out learning Activity 10.1 on page 187.
- Through class exercises, make sure each students participants in converting a real life problem or routine to an algorithm.
- After introducing the students to the class activities, direct them to the previous unit in which all the examples and activities are implemented using sequence control structure.

(2 periods)

This topic provides step-by-step procedure for writing, compiling and executing C++ programs.

Information to the teacher

Given that the learner may have no prior knowledge on how compiler works, we suggest that you use simulation or video to illustrate how C++ compiler performs the preprocess-compile-link-build process. Let the students also realize that the process can also be seen on the status bar of most compilers.

Materials: Computer, projector, C++ compiler, digital materials, internet.

Preparation

Prior to introducing the topic, ensure that you have sample program to easily distinguish high-level languages.

Teaching guidelines 10.3

- Through practical exercises, give the students more challenging activities to augment activity 10.1 on page 187 of the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Input and Output Streams

(2 periods)

This topic describes and demonstrate how C++ takes user input from keyboard though send the output to the console using the standard iostream..

Information to the teacher

The teacher should use practical examples to demonstrate differences between various iostream object and their member functions. Most beginners confuse between the insertion operation and the extraction operator >>. Therefore, it is important that you give the learner more practical exercises to help differentiate the two operators. Since he student should be made aware that more than one paradigm may be supported by a programming language. Given that C++ is object oriented, let the student also understand the concepts relating to object-oriented: class, object, function and member function.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate use of iostream.

Teaching guidelines 10.4

- Through demonstration, take the

student through examples and activities in the student book to help them understand how to manipulate inputstreams and output screams using C++ cin and cout objects

- Individually or in groups, let the learners work on program examples provided in the student's book on page. Refer the students to activity 10.2 on page 189 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Variables and Data types

(2 periods)

This topic serves as an introduction to memory allocation in programming context using variables.

Information to the teacher

To deliver this topic, we suggest that teacher make use of teaching aids such as wall simulation or wall chart let the learner visualize how memory allocation is determined C++ compiler depending on data type. This is important because the student will appreciate why C++ is a system programming language used to develop operating systems. This

approach is also very important because students can use address of operator (&) to display size of memory assigned to variables according to declared data type.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate use of variables.

Teaching guidelines 10.5

- Through demonstration, take the student through examples and activities in the student book to help them demonstrate skills in declaring and initializing variables using data types supported by C++.
- Individually or in groups, let the learners work on activities 10.3 to 10.7 (page 190-196) in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Constants

(2 periods)

This topic serves as an introduction to memory allocation in programming context, but in this case to data that does not change.

Information to the teacher

To deliver this topic, we suggest that teacher provokes the learner to explain the how constants are used in mathematics and physics. This is important because the student will appreciate how the C++ treats constant the same we understand it in the related mathematics. It is from this activity you can use sample program to show that constant do not change during program execution,

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate use of constants.

Teaching guidelines 10.6

- Through demonstration, take the student through examples and activities in the student book to help them understand how to convert a binary number into other number systems.

- Individually or in groups, let the learners work on activities 10.8 on page 197 in the student's book.

If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Output Formatting

(1 periods)

This topic introduces the student to output formatting in C++ using library `iostream` and library manipulation functions.

Information to the teacher

The teacher should use practical examples to demonstrate how C++ uses `iostream` functions and the `<iomanip>` library function to format numbers, characters and strings. The teacher should clear the air on the differences between `iostream` formatting functions and those accessed via `iomanip` interface. Give the student more practical exercise that also make use of escape sequence characters used to format literal strings and output.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

This section has a lot of computations

required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 10.7

- Through demonstration, take the student through examples and activities in the student book to help them format numeric and character/string output.
- Individually or in groups, let the learners work on activity 10.9 on page 201.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for Unit 10

Assessment Exercise 10.1 (page 184)

1. Distinctions between C and C++ in syntax, semantics, data types and programming paradigms are provided in the student book on page 170-171.
2. C++ is largely a superset of C because whatever you can do in C, you can do in C++ but not everything you do in C++ can be done in C.
3. In 1980, Bjarne Stroustrup

incorporated Smalltalk features into C with classes. Later 1984, C with classes was enhanced and renamed C++.

4. Because it supports imperative, OOP, and functional programming.

Assessment Exercise 10.2 (page 187)

1. Preprocessor and Linker
 - (a) Preprocessor directives prepares the source code for compilation by removing comments, and white spaces
 - (b) Linker combines the object code compiled from your source code with the code for the imported functions to produce an executable file
2. Preprocess, compile, and link process is well depicted in the student book on page 174.
3. C++ IDEs include GNU C++, Dev C++, Microsoft Visual C++, CodeLite, NetBeans and Eclipse..
4. Error in header file iostreams instead of iostream, no main () function, illegal use of indirection operators.
5. Program that displays Rwanda is a beautiful country:

```
#include <iostream>
using namespace std;
int main(){
cout<<"Rwanda is Beautiful
Country";
return 0;
```

```
}
```

Assessment Exercise 10.3 (page 196)

1. `char cities[30] = {"Kigali"}.`
2. C++ program for computing hypotenuse


```
#include <iostream>
#include <cmath>
int area, perimeter; //global variables
using namespace std;
int main() {
    int a, b; //local variables
    double hypo;
    cout<< "Enter side A: ";
    cin>>a;
    cout<< "Enter side B : ";
    cin>>b;
    //calculate the area and preimeter
    hypo = sqrt((a*a) + (b*b));
    //display hypotenuse
    cout<< "Hypotenuse is " << hypo<<
endl;
    return 0;
}
```

Assessment Exercise 10.4 (page 201)

1. Fixed format specifies maximum number of digits to displayed while scientific specifier displays a number in exponential form .
2. Requires use of `setw()` and `setprecision()` as follows:
 - `setprecision(2)<<fixed<<1234.56789;`
 - `setprecision(3)<<scientific<<1234.56789`
3. Newline, backspace, audio, and tabs.

Guideline to Activities

Activity 10.1: Compiling and executing C++ program (page 187)

This activity requires the learners to use skills gained in writing Hello World program to write a program similar to the one shown below:

```
#include <iostream>
using namespace std;
int main() {
cout<<"Programming in C++ is Fun"<<endl;
return 0;
}
```

Activity 10.2: Input streams and output streams (page 189)

This activity requires the learners to translate the pseudocode given on page 189 to a C++ program similar to the one shown below:

```
#include <iostream>
using namespace std;
int main() {
int x, y, z, result; // declare 4 variables
cout << "Please enter variable x:"; // input
message
cin >> x; // read x from keyboard
cout << "Please enter variable y:";
cin >> y; // read y from keyboard
cout << "Please enter variable z:";
cin >> z; // read z from keyboard
result = x+2*(y-z);
cout << "The result is:" << result << endl;
return 0;
}
```

Activity 10.3: Variables (page 190)

This is an analytical problem solving challenge that requires the learners to compare human mental retention capacity and that of computer memory.

Activity 10.4: Rules of naming variables (page 192)

This activity requires the learners to understand and explore further on reserved words provided on page 190 and 184 of the student's book.

Activity 10.5: Data types (page 193)

This activity requires the learners to appreciate the importance of declaring and using the correct data type to avoid runtime error. Memory overflow means that the number entered cannot be accommodated in the declared memory size. For example, you cannot enter a double that requires 8 bytes into a variable declared as char that reserves only 1 byte.

Activity 10.6: Declaration of variables (page 194)

1. This activity tests learner's problem solving ability to identify and replace declaration of product with long or double to avoid memory overflow.
2. The second activity requires the learner to convert the algorithm on mental sums to a program shown below:


```

#include <iostream>
using namespace std;
int main () {
int sum,diff;
int x=5, y=2;
x = x+1;
sum = x+y;
diff = sum-4;
cout<<"Sum is:"<<sum<<endl;
cout<<"Difference is:"<<diff<<endl;
return 0;
}

```

Activity 10.7: Initialization of variables (page 196)

1. This activity requires the learners to use array of characters discussed later in unit 14 to declare and use strings.
2. This activity test the learner's ability to transfer knowledge learnt in mathematics to write a the program that calculates hypotenuse using the following expression:

$$\text{hypo} = \text{sqrt} ((a*a) +(b*b))$$

Activity 10.8: Declaration of constants (page 198)

This activity requires the learners to declare pi as constant in a program that calculates area and volume of a sphere using the following declaration as assignment statements:

```

//constant declaration
const double PI = 3.142

```

```

//assignment statements
area = 4 * PI * r * r;
volume = (4/3) * PI * r * r * r;

```

Activity 10.9: Formatted output (page 201)

This activity requires the learners to create a program that calculates Body Mass Index (BMI) and display formatted body mass index (BMI). Below is a sample program the students are expected to provide:

```

#include <iostream>
#include <iomanip>
using namespace std;
int main() {
double bmi, weight, height; // declare 4 variables
cout << "Please enter weight in kgs:// input message
cin >> weight; // read x from keyboard
cout << "Please enter height in metres:";
cin >> height; // read y from keyboard
bmi = weight/(height*height);
cout<<setprecision(3)<<fixed<<bmi;
cout << "Your BMI is:" << bmi<< endl;
return 0;
}

```

Unit Test 10 (page 201)

1. Reserved words have a special meaning in a language and can only be used for intended purpose.
2. C++ supports both procedural programming and object-oriented.
3. In 1980, Bjarne Stroustrup incorporated Smalltalk features into C with classes. Later 1984, C with

classes was enhanced and renamed to C++.

4. Similarities between C and C++ in syntax, semantics, data types and programming paradigms can be obtained from internet.

5. In procedural programming, a program is broken down into procedures while in OOP, procedures and data are encapsulated into an object.

6. *Rules of choosing variable names*

- Case sensitivity: identifier written in uppercase is not the same as that written in lowercase letters.
- White spaces, punctuation marks or symbols cannot be part of an identifier.
- Only letters, digits and single underscore characters are valid; in no case they can begin with a numeric digit.
- Variable identifiers must be with a letter. For example, "3houses" is invalid.
- Identifiers cannot be a keyword or reserved by a compiler's for specific purpose.
- Avoid meaningless identifiers such as J23qrsnf.

7. Because keywords are reserved by a compiler for specific purpose and cannot be redefined.

8. This requires the student to understand the basic structure demonstrated by hello world in the student book on page 183.

9. program to calculate sum and mean of 3 numbers

```
#include <iostream>
using namespace std;
int main() {
int a, b, c, sum;
double mean;
cout << "Enter 1st number:"; cin >> a;
cout << "Enter 2nd number:"; cin >> b;
cout << "Enter 3rd number:"; cin >> c;
sum = a + b + c;
mean = sum/3;
cout << "Sum is:" << sum << endl;
cout << "Mean is:" << mean << endl;
return 0;
}
```

10. This program is similar to that of number 9 only that the student is expected to use format specifiers like set(w), setprecision() and fixed.

11. This program requires use of looping or variable for each day then calculates degrees in fahrenheit using:

$$\bullet \text{ fahrenheit} = 32 + (9 * \text{cel}) / 5$$

12. This program require use of the formula:

$$\bullet \text{ amount} = P * (1 + 0.08)^3$$

- P is the principal amount while amount is calculated value of deposit after 3 years.

PROGRAMMING

Procedural Programming

Unit II Expressions and Operators in C++ Language

Student's Book page 202 – 219. (12 Periods)

Key unit competence

By the end of this unit, the learner should be able to apply expressions and operators in C++ programming.

Learning objectives

Knowledge and understanding

- Use different operators in C++ program.
- Understand use of operators in expressions and their precedence.

Skills

- Utilize arithmetic, logic, assignment, relational, CAST, conditional, bitwise and comma operators in C++ program.
- Apply precedence of operators in C++ program.
- Interpret an expression in C++ program.

Attitudes and values

- Derive algorithm for a given problem and implement the solution logic into C++ programming language using operators and provide intended result

- Read and interpret a simple C++ program containing expressions and operators, and provide the intended results.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in four number systems. This include manipulating binary, decimal, octal and hexadecimal numbers using arithmetic operators

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity

that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Links to other subjects

This topic is highly linked to mathematics. This is because the learner is expected to apply knowledge and skills learned in mathematics to write correct expressions in C++.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Financial Education:** This is indirectly implied in activity 11.1 on page 214 (3) in which we have used a program that computes amount used in purchasing goods and services.

Assessment criteria

The learner should be able to apply operators in C++ language in the correct place and in proper order to get intended results.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

Though the learner may have used mathematical operators in C++ programs, this topic is aimed at detailed discussion of types of operators used to write valid C++ expressions. Note that we have deliberately rephrased the title to operators and expression different from the way it appears on the syllabus. This is because we strongly feel the learner should understand functions of an operator before writing expression. It is important to refer the student back to the unit on introduction to algorithms where operators and expressions were introduced.

Suggested teaching/ learning activities

To introduce C++ operators and expressions, use discovery methods, group discussion and practical exercises in the computer lab.

Expressions and operations

(2 periods)

This topic defines and gives a brief overview of operators and expressions in mathematics and computer science context. The section also serves as footprint to the other two sections that

covering operators, and expressions in details.

Information to the teacher

This unit introduces the learner to operators and expressions from mathematics point of view using arithmetic operators. This is important because students will be able to relate operators used in C++ with those used in mathematics. Curious students may ask why multiplication and division symbols are not similar to those used in mathematics. In such a case let the student the Apart from appreciating various number base systems; draw the attention of the students to standard ASCII characters supported by C++. For example, using x for multiplication is interpreted as letter x by C++ compiler. effort to connect such numbers to real world of mathematics and physics. This

Materials: Computer, projector, C++ compiler, digital materials, internet connection. You may also use wall charts to classify operators and expressions.

Preparation

Be prepared to handle weak students who have poor background in mathematics because programming requires creativity, critical thinking and problem solving skills. You may need to give a weak learner to more specialized attention using demonstrations and practical exercises.

Teaching guidelines 11.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out research activities to familiarize themselves with types of operators supported by C++. Let the group leader give each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation to help the learners appreciate importance of mathematics in computer programming.

Classification of C++ Operators

(5 periods)

This topic delves into detailed discussion of operators supported in C++ starting with familiar arithmetic operators.

Information to teacher

In this topic, we demonstrate how to categorize and use operators supported by C++. We point out that

C++ designer choose to use special symbols to represent operators because this symbols are available on standard keyboard. However, it is important to causation the student on illegal use of these characters because this may violate memory integrity. Use practical examples when covering each category of operators discussed in the student's book. Using practical exercises let the students distinguish between unary operators and binary operators.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate use of unary and binary operators.

Teaching guidelines 11.2

- Capture the attention of the class by making them carry out learning Activities 11.1 to 11.7 from page 205 to 211.
- Through class exercises, make sure each students participants classifying and demonstrating use of various arithmetic operators.
- After introducing the students to the class activities, direct them to the previous unit in which all

the examples and activities are implemented using sequence control structure.

Classification of C++ Expressions

(5 periods)

In this topic, we demonstrate how to categorize and write expressions in C++ programs. Note that in the student book, we classify expression according to number of operands, and side-effect produced by the expression. By side-effect we mean that the expression can change (mutate) the state of the target cell address. However, it is important to use more practical exercises to help the learner easily understand how to write an equation that returns desired results.

Materials: Computer, projector, C++ compiler, digital materials, internet connection.

Preparation

Prior to introducing the topic, ensure that you have a computer programming language installed in the lab. Make sure you create and test sample programs that demonstrate use of unary and binary operators.

Teaching guidelines 11.3

- Through practical exercises, give the learner more challenging problems to supplement that provided in activity 11.8 on page 217.
- If you are not satisfied with the student's answers, it would be

important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for Unit II

Assessment Exercise 11.1 (page 214)

1. Definition of terms:
 - (a) Expression refers to a sequence of operators and operands that specifies relational or mathematical computation
 - (b) Operand is numeric value manipulated by an operator
 - (c) Operator is a symbol that instructs a compiler to evaluate an expressions
2. Prefix operator is a unary that evaluates the operand on its right while a postfix evaluates an operand on its left.
3. Categories of operators include arithmetic, assignment, relational, logical, bitwise and compound operators
4. To make C++ more efficient and universally acceptable
5. The statement implies that the value of x is stored in main hence replacing y with 8
6. $y = a*(x*x*x) + b*x + 7;$
7. (a) AND=1110011, OR =1111111
(b) AND=10000, OR =1111011

(c) AND=10000010, OR =11111011

8. Arithmetic operators have higher precedence than relational operators

Assessment Exercise 11.2 (page 217)

1. Statement is more than expression consisting of comments, input, output and expressions while expression is a combination of operands and operators
2. Unary operator like ++x takes only one operand while binary operator works on two operands like x+y
3. The output is 35 for product, and 4 (bytes) returned by sizeof() operator

Guideline to Activities

Activity 11.1: Precedence rule (page 205)

1. This activity requires the learners to use precedence rule in arithmetic expressions that returns the value of x as: 786
2. If the precedence rule is not applied, the value of x would be different from 786. Assuming evaluation starts from left to right, the value returned would be something like 278.333.
3. This activity requires the learner to identify the assignment statement:
amount = quantity*price;
4. The assignment statement expected is:
 $a*(x * x) + b * x + c;$

Activity 11.2: Assignment operator (page 205)

This activity requires the learner to determine output produced after replacing variables a and b with 12 and 15 respectively. This should produce a=>15; b=> 5; c => 25.

Activity 11.3: Increment and decrement operators (page 207)

This activity tests learner's ability to use C++ increment and decrementation can be done like this: orange-- (14) ; banana-- (34); isombe-- (12); clients--. This should produce orange++ (16) ; banana++ (36)5; isombe++ (14); clients++ (4).

The decrement operation can be done like this:

orange-- (14)
banana-- (34)
isombe--(12)
clients-- (2).

Activity 11.4: Relational operator (page 208)

This activity tests learner's ability to determine the outcome of Boolean expression. The expected answers for each conditional logic is as follows: (x= y) =>0; (x>y) =>1; (x != y) =>0; (x<y) =>0;.

Activity 11.5: Logical operators (page 209)

This activity tests learner's ability to

determine the outcome of Boolean expression. The expected answer for statement logic is as follows:

- (x<=35) && (z==24) =>0
- (x= =35) || (y<10)=>1;
- (x>y) && (y<z)=>1;

Activity 11.6: Bitwise operators (page 210)

$p \& q$	$p q$	$p \wedge q$
0	0	0
0	1	1
1	1	0
0	1	1

Activity 11.7: Conditional operators (page 211)

This activity tests learner's ability to determine the outcome of selection based on the following conditional logic.

- 7= =5?4:3 =>3
- 7>=5+2?4:3 =>4 ;
- 5>3?a:b =>a;
- a>b?a:b =>b;

Activity 11.8: Expressions and operators (page 217)

1. This is an exploratory activity that requires the learner to identify different meaning (semantics) of the + and – symbol => binary operator, unary operator, increment/ decrement operator.

2. Binary expressions have two operands and manipulated by an operator between them while ternary expression takes three operands and two operators (?:).

Unit Test II (page 218)

1. Definition of terms:

(a) Operator precedence is priority assigned to each operator.

(b) Self-assigned operators have uniquely way of combining arithmetic and assignment operators

2. Prefix operator is a unary that

evaluates the operand on its right while a postfix evaluates an operand on its left.

3. Makes C++ more universal, efficient, and easier to learn

4. Inclusive OR check if two conditions are both true or either must be true for the return to be true. Exclusive OR means that one of the two conditions must be true and the other false

5. The sizeof() returns memory size allocated to a variable in bytes e.g. 4, while & returns actual address.

6. Requires understanding of ASCII character set e.g. 97 for lowercase a and 65 for uppercase A

7. See operator precedence summary on page 212 in the student's book.

8. `#include<iostream>`

`using namespace std;`

`int main(){`

`const double pi = 3.141596;`

`double radius, volume, area = 0.0;`

`cout << "Please enter the radius of sphere" << endl;`

`cin >> radius;`

`volume = ((4.0 / 3.0) * pi * radius*radius*radius);`

`area = 4.0 * pi * radius*radius;`

`cout << "The volume area is: "<< volume << endl;`

`cout << "The surface area is: "<< area << endl;`

`return 0;`

`}`

9. `#include <iostream>`

`#include <cmath> //for using power and root functions`

`using namespace std;`

`int main() {`


```

double a =2, b=6, c=4;
double root1, root2, discriminant;
discriminant = (pow(b,2) - 4*a*c);
root1 = (((-b) + sqrt(discriminant))/(2*a));
root2 = (((-b) - sqrt(discriminant))/(2*a));
if (discriminant == 0) {
cout <<"The equation has same roots."<<root1<<" "<<endl;
}
else if (discriminant > 0) {
cout<<"The discriminant is " <<discriminant<<endl;
cout<<"The two roots are:"<< root1<<" "<<root2<<endl;
}
else {
cout <<"The equation has complex roots."<<endl;
}
return 0;
}

```

10.

```

D:\Rwanda4\C++Programs\ExtraPrograms>
Enter first number: 75
Enter second number: 78
num1 = 75; num2 = 78
num1 + num2 = 153
num1 - num2 = -3
num1 * num2 = 5850
num1 / num2 = 0.961538

```

11. The errors to be corrected are: declaration of num1 and num2 to double or float, replace >> with << in the first, third and fifth cout statements, declaration of dec2, replace \ with division operator / in the last two cout statements, replace end with endl, and insert return 0 before ending main. Below is the corrected code:

```

#include <iostream>
using namespace std;
int main() {
int dec1 = 2, dec2 = 4;
double num1 = 2.5, num2 = 5.0;
cout << dec1 << " + " << dec2 << " = " << dec1+dec2<<endl;
cout << num1 << " + " << num2 << " = " << num1+num2 << endl;
cout << dec1 << " - " << dec2 << " = " << dec1-dec2<<endl;
cout << num1 << " - " << num2 << " = " << num1-num2 << endl;
cout << dec1 << " / " << dec2 << " = " << dec1/dec2<<endl;
cout << num1 << " / " << num2 << " = " << num1/num2 << endl;
return 0;
} //end main

```

Programming

Procedural Programming

Unit 12

Control Statements in C++

Student's Book page 220 – 242 (16 Periods)

Key unit competence

By the end of this unit, the learner should be able to use control statements in C++ program to implement branching and iterations.

Learning objectives

Knowledge and understanding

Describe various control statements used to implement conditions and iterations in C++ program.

Skills

Apply control statements in C++ program.

Attitudes and values

Derive algorithm for a given problem and implement the solution logic into C++ programming language using operators in correct place and in correct order.

Read and interpret a simple C++ program containing conditional branching and looping statements, and provide the intended results.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes

on computing accuracy in computer programming.

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Genocide studies: This unit addresses this emotive issue by providing a selection structure that uses looping control statement to display “Genocide Never!” five times on the screen.

Links to other subjects

Most of the concepts in this unit are linked to mathematics, more specifically arithmetic and Boolean algebra. This is because the learner is expected to apply arithmetic and algebraic skills in designing control statements.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In this unit, we have addressed the issue of standardization through use of standard C++ language library.
3. **Financial Education:** several activities and assessment questions have been used to demonstrate prudent financial management.

Assessment criteria

The students should be able to use control statements in C++ program to effectively implement conditional branching and iterations. Apart from activities and exercises provided in the student book, the teacher should use other assessment methods and tools to test whether the learner has acquired necessary knowledge, skills and attitude in solving problems that require use of control statements in C++.

Suggested teaching methodology

Guided discovery

Research

Question and answer

Discussion

Background information

This unit builds on computer algorithms to equip the learner with knowledge and skills on how to design algorithms using condition logic implemented using selection and looping control structures. To test the logic of each control structure illustrated in the student’s book, we recommend use of trace tables and programming languages. Similar to approach used in the previous unit, we make no reference to specific programming language.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, group discussions and practical programming to implement control structure earlier discussed in

Sequence Control Structure

(2 periods)

This section contains an overview sequence control. The section also serves as a footprint to sections that demonstrate use of conditional logic in selection and looping statements discussed later.

Information to the teacher

To widen the learner's understanding of sequential control structure, we suggest that the teacher provide more activities and exercises to complement what is provided in the student's book. Where necessary, practical demonstration should be used to help the learners understand difficult concepts. Your approach should make the learner appreciate design of sequential algorithms covered in unit 7.

Materials: Computer, projector, C++ compiler and digital materials. You may also use wall charts depicting sequential control structure.

Preparation

Be prepared to handle weak students who have poor background in mathematics because use of control statements requires creativity, critical thinking and problem solving skills.

Teaching guidelines 12.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups

and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.

- Ask the groups to carry out Activity 12.1 on page 221 in the student's book. Let the group leader give each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should demonstrate to the problem in a practical session.
- Provide summary from practical work in order to help the learners understand that how sequential execution is performed in computer programs.

Selection Statements in C++

(2 periods)

This topic provides a more detailed discussion of selection control statements starting with if construct, the proceeding to if...else, nested if, and switch.

Information to the teacher

The teacher should demonstrate how to implement selection control statements using C++ examples. As a class or group activity, let the students compare nested if selection and switch statements. We encourage the teacher to use brainstorming sessions to guide students in identifying situations that demand use

of each of the four selection statements in C++.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of selection statements.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various selection options.

Teaching guidelines 12.2

- Through demonstration, take the student through examples and activities and in the student book to help them demonstrate use of selection control statements in C++.
- Individually or in groups, let the learners work on activities 12.2 to 12.5 from page 223 to 228 in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Looping Statements in C++

(4 periods)

This topic provides detailed discussion of looping statements starting with for loop. The teacher should then demonstrate how to implement loops in

C++ programs. Let the learner research on internet or relevant resource material to clearly compare the behaviour of the while, do...while and nested loops. We also encourage the teacher to use brainstorming sessions to guide students in identifying situations that demand use of the looping, break and continue statements.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of looping statements.

Preparation

This section has a lot of computations required. Let the students prepare well to make sure that they perform correct octal to decimal computations. Sensitize them on the need to have electronic calculators to simplify some computations.

Teaching guidelines 12.3

- Through demonstration, take the student through examples and activities in the student book to help them understand and use the four looping controls in C++ programs.
- Individually or in groups, let the learners work on activities 12.6 on page 228 and 12.7 on page 238.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for unit 12

Assessment Exercise 12.1 (page 228)

1. Selection refers to a situation where further program execution depend one making a decision from two more alternatives.
2. Four selection control structures are: if, if..else, nested if and switch.
3. Nested if is used to test multiple options e.g. in assigning grades same as switch selection
4. When more than one alternatives id to be selected, a decision has to be made based on conditional logic
5. simplicity, available options, data type

```
#include <iostream>
using namespace std;
int main() {
    int score;
    cout << "Enter mean score:";
    cin>>score;
    if (score>= 50) {
        cout<<"Pass\n";
    }
    return 0;
}
```

Assessment Exercise 12.2 (page 237)

1. Looping refers to repeated execution of a statement until a certain condition is fulfilled while selection refers to taking alternative action.
2. Looping controls in C++ are while, for and do..while

3. While is a pre-test loop that executes statement within the loop at least zero times while do..while is post-test hence it executes statements at least once.
4. Saves coding time, more clear, saves compile time.
5. This questions require students to use the following formats:

```
do {
    statements;
} while(condition);

for(initial;condition;
step){
    statements;
}
```

The student should be able to step in a for loop such as shown below:

```
#include <iostream>
using namespace std;
int main () {
    for (int count=1; count<=200;
count=count+2) {
        cout << count << " , ";
    }
    return 0;
}
```

Assessment Exercise 12.3 (page 241)

1. Jumps supported by C++ are goto, exit(), continue, an break jumps.
2. Indiscriminate use of the statement can be a source of program bugs that may be hard to detect and debug
3. Break exits from the loop or selection

while control cause the program to skip the next statements if the condition returns true.

4. Exit() is a function that terminates a program prematurely
5. Exit can be used to exit from a loop or program.
6. See Student's Book on page 217 and 218 programs that demonstrates use of continue and goto jumps.

Guideline to Activities

Activity 12.1: Sequence Control Structure (page 221)

This activity requires the learners to demonstrate understanding of sequence structure by writing a program swaps three numbers. Refer the learner to sample program on pages 218 to 219 that demonstrates how to swap user input.

Activity 12.2: if selection statement (page 223)

Below is a sample program the learner is expected to provide to write to demonstrate understanding of if selection. Refer the learner to sample program on pages 220 that demonstrates the if selection.

Activity 12.3: if ... else selection statement (page 224)

This activity requires the learners to translate the pseudocode given on page 222 to a C++ program that implements if..else selection similar to the program

shown below:

```
#include <iostream>
using namespace std;
int main() {
    int x,y;
    double result;
    cout << "Enter two numbers x:";
    cin >> x;
    cout << "Enter two numbers y:";
    cin >> y;
    if (y==0) {
        cout << "Error: Division by zero";
    } //end if
    else {
        result = (double)x/y;
        cout << "Result of x/y is: "<<result<<endl;
    } //end else
    return 0;
} //end main
```

Activity 12.4: Nested if selection statement (page 226)

This activity test the learner's ability to use of nested if selection in a real life scenario. Assist the student apply the concepts learnt in the example on assigning grade but they may have to use string type (string object) and initialize medal to nil. The following is a program that assign medals to athletes:

```
#include <iostream>
using namespace std;
int main() {
    int position;
    string medal= "Nil";
```

```

cout << "Enter the runner's position:";
cin>>position;
if (position >0) { //ensure user input is non-
zero
    if (position == 1) {
        medal="Gold";
    }
    else if (position == 2) {
        medal= "silver";
    }
    else if (position == 3) {
        medal = "bronze";
    }
    else {
        medal = "Thanks for your participation";
    }
} //end outer if
else { //trap a non-zero user input
    cout<<"Position must be 1 and above\n";
}cout << "Medal award is:"<< medal<<endl;
return 0;
}

```

Activity 12.5: Switch selection statement (page 228)

This activity test the learner's ability to use of the nested if selection in a real life scenario. You may have to assist the learners in replacing the segment of the inner if..else..if demonstrated in Activity 12.4 on page above.

Activity 12.6: Looping control structure (page 228)

This activity test the learner's ability to use of looping control statements

in a real life scenario. The following is a sample program:

```

#include <iostream>
#include <iomanip>
using namespace std;
int main () {
    int count=1, sum=0, number;
    double average;
    while (count <=5) {
        cout << "Enter a number:";
        cin >> number;
        sum = sum+number;
        count++;
    } //end while loop
    average = (double)sum/count;
    cout<<setprecision(2)<<fixed<<average<<endl;
    cout<<"The sum is: "<<sum <<endl;
    cout<<"The average is: "<<average <<endl;
    return 0;
}

```

Activity 12.7: Looping control statements (page 238)

This activity test the learner's ability to use for loop and break statements. The following is a sample program implemented using for loop within which is the if selection that causes a break if the number is zero and below:

```

#include <iostream>
#include <iomanip>
using namespace std;
int main(){
    int number, sum=0, count=0;
    double average;
    while (count <20) {
        cout << "Enter a number:";

```



```

cin >> number;
if (number >0) {
    sum = sum+number;
    count++;
} //end if
else {
    cout <<"loop exits at:"<<count<<endl;
    break;
} //end else
} //end while
average = (double)sum/count;
cout<<setprecision(2)<<fixed<<average<<
endl;
cout<<"Cummulative total: "<<sum <<endl;
cout<<"Average: "<<average <<endl;
return 0;
}

```

Activity 12.8: Break, continue and exit() (page 241)

1. The first activity test the learner's ability to use for loop and break statements as demonstrated in activity 12.7.
2. The second activity test the learner's ability to use for loop, break and continue statements as demonstrated in activity 12.7. However, in this case, the student should trap entries above 100 using continue to allow the next iteration to be executed.
3. The third activity test the learner's ability to use loop, exit and output as demonstrated in the following program:

```

#include <iostream>
using namespace std;
int main(){

```

```

int i,j,rows,k=0;
cout<<"Enter number of rows: ";
cin>>rows;
for(i=1;i<=rows;i++) { //rows
    for(j=1;j<=rows-i;j++) { //cols
        cout<<" ";
    } //end inner for loop
    while(k!=2*i-1){
        cout<<"* ";
        k++;
    } //end while loop
    k=0;
    cout<<"\n"; //line breaks
} //end outer for loop
return 0;
} //end main

```

Unit Test 12 (page 241)

1. If does not provide alternative for force while if---else does.
2. Nested for and do..while structures
 - a) Nested for- provide a nested loop for for, while or do..while.
 - b) do..while provide an example of post-test besides those provided in the student book
3. Requires initializing count to 50 and decrement the count by 1 as follows:

```

count = 50;
while(count>=50){
    cout<< count<< " , ";
    count--
}

```
4. The program require use of nested if or switch selection control statements
5. Requires use of control statement to

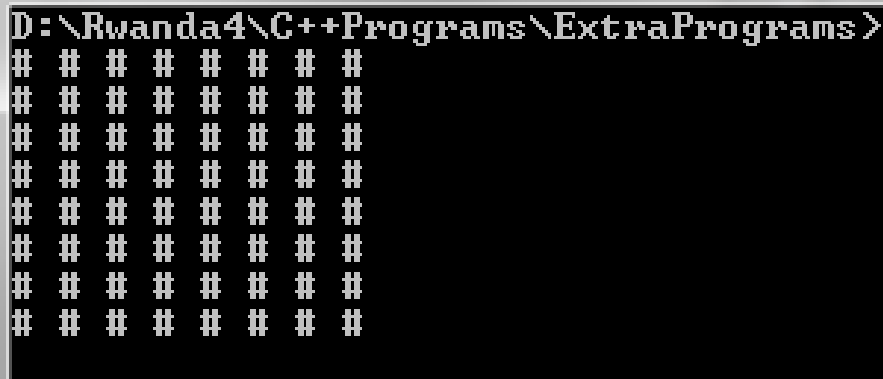
accept n as input, increment sum that in initially assigned to zero as follows:

```
#include <iostream>
using namespace std;
int main() {
int count =0, maxcount, number;
double average; int sum =0;
cout<<"How many numbers";
cin>>maxcount;
while (count<maxcount){
cout<<"Enter a number";
cin>>number;
sum=sum+number;
count++;
}
average=(double)sum/count;
cout<<"Sum "<<sum<<endl;
cout<<"Average "<<average<<endl;
```

}

6. Requires use of the following expression to convert celcius to degrees
 $fahrenheit = 32 + (9 * celcius) / 5$
7. Require use of the following expression to computer interest:
 $amount = principal * (1.08)^t$
8. Require use of the following expression to computer interest:
 $amount = principal * (1.12)^t$
9. Should be avoided or limited to use in sequential programs when absolutely

10nec



```
D:\Rwanda4\C++Programs\ExtraPrograms>
#####
#####
#####
#####
#####
#####
#####
#####
#####
```

Programming

Procedural Programming

Unit 13

Functions in C++ Programming

Student's Book page 243– 261. (14 Periods)

Key unit competence

By the end of this unit, the learner should be able to define and use functions in C++ program.

Learning objectives

Knowledge and understanding

- Describe different predefined functions in C++ programming language.
- Describe the steps of using functions in C++ program.

Skills

- Define a function in C++ language
- Declare a function in C++ language
- Call a function in C++ language

Attitudes and values

- Appreciate the importance of functions for reusability and modular design.
- Solve a given algorithm and apply the solution into C++ programming language using function.
- Read and interpret a simple C++ program containing function and provide intended result.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in computer programming.

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner

working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Links to other subjects

Most of the concepts in this unit are linked to mathematics. This is because the learner is expected to apply arithmetic and algebraic skills in defining functions.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Financial Education:** Activities such as 13.1 (2) on page 249 and assessments exercise 13.1 (9), and Unit Test (9) have been used to analyze investment in the banking sector.

Assessment criteria

The students should be able to define and use functions in C++ program. Apart from activities and exercises provided in the student book, the teacher should use other assessment methods and tools to test whether the learner has acquired

necessary knowledge, skills and attitude in solving problems that require use of C++ functions.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

This unit introduces the learner to modular programming using C++ library and user-defined functions. To test the logic of functions demonstrated in the student's book, we recommend use of trace tables.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, question and answer method, group discussions and practical programming to implement procedural programming using C++ language.

Fundamentals of C++ Functions

(1 period)

This section contains an overview of top-down approach to modular programming using C++ functions. The section also serves as a footprint to sections that demonstrate use of library and user-defined functions in procedural programming.

Information to the teacher

To widen the learner's understanding of functions in C++, we suggest that the teacher provide activities and exercises to drive home concepts provided in this section. Where necessary, use simple programs containing library and user-defined functions and let the learner to interpret them.

Materials: Computer, projector, C++ compiler and digital materials. You may also use wall charts depicting top-down approach to modular programming.

Preparation

Be prepared to handle weak students who have poor background in mathematics because use of functions requires creativity, critical thinking and problem solving skills.

Teaching guidelines 13.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to brainstorm on programming paradigm supported by functions in C++. Let the group leader give each member an opportunity to give description and solution to the problem using top-down approach.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from

their presentation in order to help the learners understand that selection and control structures are instrumental in day-to-day's problem solving.

Types of functions: Library Functions

(5 periods)

This discusses in details type of functions starting with C++ library functions used to manipulate numeric data, characters and strings.

Information to the teacher

The teacher should demonstrate how to use C++ library functions: math, character and string library functions provided in the student's book. Once the students have mastered use of library function, introduce them to more advanced applications of the functions in real world scenarios such as computing loan repayment problem in activity 13.1 on page 149. It is important to emphasize the return type using examples of function prototypes available in C++ standard documentations.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of selection statements.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various selection options.

Teaching guidelines 13.2

- Through demonstration, take the student through examples and activities and in the student book to help them understand how functions work
- Individually or in groups, let the learners work on exercises and activities provided in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Types of functions: user-defined functions

(4 periods)

This topic provides the learner with knowledge and skills required to create user-defined functions.

Information to the teacher

The teacher should then demonstrate how to create user-defined function in C++ programs. Let the learner research on internet or relevant resource material to clearly distinguish between monolithic programs and modular programs. We also encourage the teacher to use more sessions to guide students on parameter passing by value because most beginners find it hard to conceptualize parameters passing by reference. It is also important

that you emphasize on convention of declaring functions using function prototype (signature) before the main() function, and minimal declaration of global variables.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of looping statements.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various function return types.

Teaching guidelines 13.3

- Through demonstration, take the student through examples and activity 13.2 on page 252 in the student book. let the students appreciate benefits of modular programming
- Individually or in groups, let the learners work on activities in the student's book.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Function declaration

(1 periods)

This discusses in details of function

declaration in terms of arguments and return type.

Information to the teacher

The teacher should demonstrate how to declare or define functions that return simple data or void using different examples. Once the students have how to declare a function use of provide practical exercises to emphasize on rules of defining parameters list and the return type.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of various return types and parameter list.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various selection options.

Teaching guidelines 13.4

- Through demonstration, take the student through examples and activities and in the student book to help them understand how functions work
- Individually or in groups, let the students work on activities 13.3 to 13.5 from page 253 to 256 in the student book. It is be important to set aside time for more practical exercises to assist weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Recursive functions

(3 periods)

This topic provides detailed discussion of special type of functions that calls itself hence the term recursive function.

Information to the teacher

Recursive functions are used to implement looping in functional programming paradigm. Let the student appreciate the fact that C++ also implements functional programming using recursion but not the library or user-defined functions that we have covered earlier. Demonstrate how to implement recursive functions by stating the base case. It is advisable to let the students work in group to implement several recursive functions beyond those provided in the student's book.

Materials: Computer, projector, C++ compiler, and chart showing general syntax of looping statements.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various examples of recursive functions.

Teaching guidelines 13.5

- Through demonstration, take the student through examples and activities in the student book to help them understand how recursive functions works
- Individually or in groups, let the learners work on activity 13.6 on page 257 as well as activity 13.7 on page 259 in the student's book.

It is advisable to evaluate each learner's understanding of the concepts on modular programming before closing the unit or proceeding to the next unit.

Answers for unit 13

Assessment Exercise 13.1 (page 256)

1. Function definition contains implementation of its body while declaration does not.
2. Advantages and disadvantages of using functions
Advantages
 - A modular program is easier to understand
 - Easier to debug or modify a modular program
 - Increases programmer productivity
 - Libraries can be used in other programs*Disadvantages*
 - Requires more memory and processor time
 - Integration of various modules may be difficult
 - Global sharing of data may be dangerous
3. Five characteristic of functions:
 - Complete subprogram
 - Perform well defined task
 - Can be compiled and tested independently
 - Has only one entry and one exit
 - Can be used in different programs
4. $y = a * (\text{pow}(x, 3) + b * (\text{pow}(x, 2) + c * x + d)$
5. Character functions include: `tolower()`, `toupper()`, `isupper()`, `isdigit()`, `isalpha()`.
6. Both void and empty parenthesis means a function does not receive parameters.

7. Global identifiers are accessible by functions in the scope while local are only accessible within a function.
8. Parameter passing is well illustrated in the student's book page 252.
9. Although the problem is similar to other problem of calculating amount. This requires use of function that receives parameters from `main()`:

Assessment Exercise 13.2 (page 260)

1. *Recursion and recursive functions*
 - (a) Recursion is the process of repeating items in similar manner
 - (b) Recursive function is a function that calls itself in a similar manner directly or indirectly.
2. Recursion is tested against a base case while iteration is tested against loop variable.
3. Infinite recursive function:
 - (a) He might not have defined the base case.
 - (b) He need to use the if selection to test the base case against each recursion .
4. Recursive functions can be expensive in both processor time and memory space
5. Students are expected to use the sample code provided on page 256 to implement a program that returns GCD as shown by the sample program below:

```
#include<iostream>
using namespace std;
int gcd(int x,int y){
```



```

if ( y == 0 )
return x;
else if ( x == 0 )
return y;
else
return gcd(y, x%y);
}
main() {
int j,k,gcd_jk;
cout<<"Enter two numbers:";
cin>>j>>k;
gcd_jk=gcd(j,k);
cout<<"GCD is: " <<gcd_jk<<endl;
return 0;
}

```

Guideline to Activities

Activity 13.1: Library functions (page 249)

1. This activity test the learner's ability to identify and use library functions such as pow(), sqrt() and log() listed in the student's book on page 242.
2. This activity test the learner's ability to transfer knowledge learnt in mathematics or economics to calculate loan repayment using the formula:

$$P = r(PV)/1-(1+r)^{-n}$$

Where p = payment; PV = present loan value; r =rate per period (annual/12); n=number of periods (4 x 12 months).

The following is a sample program that uses pow() function to calculate monthly loan repayment in the second activity :

```

#include <iostream>
#include <cmath>
#include <iomanip>

```

```

const double RATE =0.12;//equivalent to 12%
using namespace std;
int main() {
int period =48;//4 years x 12 months
double loan,repayment, month_rate; //
cout << "Enter initial loan amount:";
cin >> loan;
month_rate =RATE/12;
//use this expression to calculate monthly loan
repaymet;
repayment=loan*month_rate/(1-
pow(1+month_rate,-period));
cout<<setprecision(2)<<fixed<<repayme
nt<< endl;
cout<<month_rate;
cout << "Your monthly loan repayment is:"<<
repayment<<endl;
return 0;
}

```

Activity 13.2: User-defined functions (page 252)

This activity test the learner's ability to identify formal parameters, and return type from the function:

- Formal parameters: x, y and z
- Return type: double
- Value returned: maxValue

Activity 13.3: Function declaration (page 253)

1. In the first activity, the learner's should clearly explain the return type void and difference between formal and actual arguments.
2. The following program is an shows implementation of grade processing program consisting of three user-defined functions: main(), calculator() and grader().

```

#include<iostream>
using namespace std;
char grade;
//declaration of calculator and grader
double calculator (int, int, int, int);
char grader (double);
int main (){
double average;
int math, comp, sci, econ;
cout << "Enter marks for maths:";
cin >> math;
cout << "Enter marks for computer:";
cin >> comp;
cout << "Enter marks for science:";
cin >> sci;
cout << "Enter marks for economics:";
cin >> econ;
average =calculator (math, comp, sci, econ);
grade =grader (average);
cout<<"Your average score is"<<average<<endl;
cout<<"Your grade is"<<grade<<endl;
return 0;
}
//calculator receives scores from main
double calculator (int math, int comp, int sci,
int econ) {
int total;
double average;
total = math+comp+sci+econ;
average = (double)total/4;
return average;
}
//grader use score to assign grade
char grader (double average) {
if ((average >= 80) && (average <= 100)){
grade = 'A';
}
else if ((average >= 65) && (average <= 79)){
grade = 'B';
}
else if ((average >= 50) && (average <= 64)){
grade = 'C';
}
else {
grade = 'F';
}
return grade;
}

```

Activity 13.4: Function return type and arguments (page 254)

1. Failing to declare return type in C++ is illegal but some C++ compilers

supports implicit return type such as int but with warning message. For example Microsoft Visual C++ compiler may return an error message:

error C4430:missing type specifier - int assumed.

Note: C++ does not support default-int

2. This activity requires the student to demonstrate parameter passing and return type by modifying the program created in activity 13.3 (2) on page 251 to use void in printGrade() function that receives parameters from the grader() function.

Activity 13.5: Parameter passing (page 256)

1. This activity test the learner's ability to transfer knowledge learnt in mathematics to write a modular program that calculates distance between two points.
2. This activity tests learner's ability to identify function arguments and return type. In this case, the function cannot return $x + y$ because its return type is void.

Activity 13.6: Recursive functions (page 257)

This activity test the learner's ability to transfer knowledge learnt in mathematics to write a recursive function such as factorial in student's book on page 254. Use the GCD function provided on page 256 to guide the students on how to solve the problem.

Activity 13.7: Recursive functions (page 259)

This activity test the learner knowledge and ability to use sample programs and code segments provided in the student's book on page 255 and 256 to generate Fibonacci series and determine GCD of two numbers.

Unit Test 13 (page 260)

1. Definition of terms
 - (a) Function is a self-contained sub-program that performs specific task
 - (b) Argument: data transferred or received by a function
 - (c) parameter passing is transfer of arguments between functions
2. Library functions are in-built into a compiler while user-defined are written by the programmer
3. requires use of an expression like $\text{volume} = \frac{4}{3} \pi r^3$;
4. Both void and empty parenthesis means a function does not receive parameters.
 - (a) She may not have defined formal parameters or return type.
 - (b) She can eliminate the bug by providing proper definition of arguments and return type.
5. It the interface or communication link between functions.
6. Global sharing of data may be cause a program to produce invalid output.
7. This program requires students to implement the sample code provided on page 243.
8. The program require the student to convert previous program for calculating temperature to a modular program with main, a function that returns double, and a third function that just prints the results.
9. This problem is similar to monolithic programs demonstrated earlier only that it requires use of three function. One of the function may use the switch selection to display menu list.

10.

```
D:\Rwanda4\C++Programs\ExtraPrograms>
The result is: 121
```

Programming

Procedural Programming

Unit 14

Arrays in C++ Programming

Student's Book page 262 – 276. (12 Periods)

Key unit competence

By the end of this unit, the learner should be able to use arrays and strings in a C++ program.

Learning objectives

Knowledge and understanding

- Describe use of arrays and strings and their memory representation.

Skills

- Define, declare and use arrays in C++ programming language.
- Define, declare and use string in C++ programming language.
- Manipulate array in a C++ program

Attitudes and values

- Appreciate the importance of arrays and string.
- Solve a given algorithm and apply the solution into C++ programming language using array and strings.
- Read and interpret a simple C++ program containing arrays and strings, and provide intended result.

Generic competences addressed in this unit

Numeracy: This unit highly emphasizes on computing accuracy in computer programming.

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of computer programs. The learner is expected to transfer these skills to programming in C++.

Creativity and Innovation: This unit is critical to creativity and innovation because the computations used helps the student explore various solutions to presented challenges.

Links to other subjects

Most of the concepts in this unit are linked to mathematics and computer architecture. This is because the learner is expected to apply arithmetic and algebraic skills in defining arrays.

Cross cutting issues addressed in this unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used neutral phrases.
2. **Standardization culture:** In activity 14.1 on page 264, we have addressed the issue of equitable access to transport through electronic booking of travel.
3. **Financial Education:** A number of exercises and activities in this unit have been used to demonstrate effective customer management in financial institution.

Assessment criteria

The students should be able to use arrays and strings in a C++ program. Apart from activities and exercises provided

in the student book, the teacher should use other assessment methods and tools to test whether the learner has acquired necessary knowledge, skills and attitude to solve problems that require use of C++ arrays.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

This unit introduces the learner to array and strings to appreciate importance of complex data structures in C++. To test the logic of arrays and functions demonstrated in the student's book, we recommend use of sample programs.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, question and answer method, group discussions and practical programming to implement one-dimensional arrays using C++ language.

One-dimensional Arrays

(1 period)

This section contains detailed implementation of one-dimensional arrays using C++ introduced earlier in Unit 8 under control structures and one dimension arrays.

Information to the teacher

To widen the learner's understanding of arrays in C++, we suggest that the teacher

provide more activities and exercises to complement what is provided in the student's book. Where necessary, use sample programs containing arrays and string data types and let the learner to interpret them.

Materials: Computer, projector, internet connection, C++ compiler and digital materials. You may also use wall charts depicting how array is stored in computer memory.

Preparation

Be prepared to handle weak students who have poor background in mathematics because use of control statements requires creativity, critical thinking and problem solving skills.

Teaching guidelines 14.1

- Organize the class into pairs or groups. Sensitize them on the importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.
- Ask the groups to carry out group activities to review concepts discussed in Unit 8. Let the group leader give each member an opportunity to give description and solution to real life problem that involves use of arrays.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners understand that selection and control structures are instrumental in day-to-day's problem solving.

(5 periods)

This topic discusses in details how to create and manipulate one dimensional array with more emphasis on numeric data.

Information to the teacher

The teacher should demonstrate how to use declare, define and initialize one dimensional array. Once the students have mastered use of provided exercises, introduce them gradually to creating arrays on their own. Depending on the level of learner, you can demonstrate how an array can be passed as parameter to a function.

Materials: Computer, projector, C++ compiler, and chart showing one dimensional array in memory.

Preparation

Make sure that you have tested and debugged sample programs you intend to use in class to demonstrate various how computer allocates memory to an array.

Teaching guidelines 14.2

- Through demonstration, take the student through examples and activities in the student book to help them understand how functions work.
- Individually or in groups, let the learners work on activities 14.1 to 14.3 from page 264 to 269.
- If you are not satisfied with the

student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of concepts presented before proceeding to the next unit.

Array of characters

(5 periods)

This topic provides the learner with knowledge and skills on how to create and manipulate string defined as one-dimension array of characters.

Information to the teacher

The teacher should demonstrate how to use declare, define and initialize one dimensional array of characters. Once the students have mastered use of provided exercises, introduce them gradually to creating and manipulating strings on their own. Depending on the level of learner, you may also demonstrate how to use various library functions designed to manipulate string and an object or array of characters.

Materials: Computer, projector, C++ compiler, and chart showing array of characters.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various function return types.

Teaching guidelines 14.3

- Through demonstration, take the student through examples and

activities in the student book to help them understand how to create and manipulate array of characters/strings.

- Individually or in groups, let the learners work on activities 14.4 on page 272 and 14.5 on page 275 respectively.
- If you are not satisfied with the student's answers, it would be important to organize for practice through remedial sessions to help the weak students. Then evaluate each learner's understanding of the concepts presented before proceeding to the next unit.

Answers for Unit 14

Assessment Exercise 14.1 (page 269)

1. One dimensional array is a list while multi-dimensional has more than one subscripts.
2. `int cashflow [99];`
3. First element is denoted by index 0 while the last is indicated using subscript 98.
4. Array initialization:
 - (a) `int passegger [6] = {64,25,69,67,80};`
 - (b) `int passager [99] = {0};`
 - (c) The requires use of for loop to write the default values as shown in student's book page 268.
5. This problem is C++ implementation of an algorithm discussed earlier in Data structures and algorithm on page 148.
6. Program error debugging
 - (a) The maximum size is 10 but the for loop runs 11 iterations causing an overflow error.
 - (b) There is no `ax[3]` because array count starts from 0.

7.

```
D:\Rwanda4\C++Programs\ExtraPrograms>
Mean is 65.57
Std deviation is:13.56
```

8. The errors to be corrected are: declaration of index, replace C with lowercase c, replace MAX and MAX_ARRAY with SIZE. Below is the corrected code that adds elements of array a and b and stores the new value in array c:

```
#include <iostream>
using namespace std;
int main (){
const int SIZE = 5;
int a[SIZE], b[SIZE],c[SIZE] ;
int index;
for (index = 0; index < SIZE; index++){
    cout << "Enter elements for array [a]:";
    cin >>a[index];
}
for (index = 0; index < SIZE; index++){
    cout << "Enter elements for array [b]:";
    cin >>b[index];
}
for (index = 0; index < SIZE; index++){
    c[index] = a[index]+ b[index];
}
for (index = 0; index < SIZE; index++){
    cout << "array a is " << a[index] << endl;
    cout << "array b is " << b[index] << endl;
    cout << "array c is " << c[index] << endl;
}
return 0;
}
```

Assessment Exercise 14.2 (page 275)

1. The character is a string terminator that indicates the end of a string.
2. char towns [30];
3. Each character of Buyoga will occupy one byte with the last byte being occupied by '\0' null character
4. char customer [30] = {"Ann", "Ben", "Helen", "Paul", "Joy", "Ken"};
5. This requires use of for loop in in both read and write functions.
6. The two are the same because they are two alternatives of initializing a string.

Guideline to Activities

Activity 14.1: Declaration of arrays (page 264)

This activity test the learner's ability to represent and declare one-dimensional array as follows:

- a) Array name=>points; data type =>int; elements => 6; array name=>temperature; data type =>double; elements => 6
 - b) The sample declaration for each of the two arrays is: `int points[6];`
`double temperature[6];`
7. This activity test the learner's ability to create a one-dimensional array. The following is a declaration for an array of 30 integers for the bus problem:
- `int bus[30] ;`

Activity 14.2: Initializing an array (page 265)

1. For this activity, expected sample array initialization is:
`int points[] = {21,32,43,54,65,76,87,88,99,200};`
2. The initialization does not cause an error because products populates the extra two elements with zeros
3. This activity test the learner's understanding one-dimensional array initialization and loop construct to determine the output such as shown in Fig. 14.2 on page 265 of the student's book:

Activity 14.3: Reading and writing array elements (page 269)

This activity test the learner's problem solving ability from the case study

provided and be able to store the responses in a one-dimensional array using the for loop for reading and another for loop for displaying the responses from the array elements. Refer the student to sample program on page 264 as hint to solving the problem.

Activity 14.4: Initializing strings (page 272)

1. This activity test the learner's ability to represent and declare one-dimensional array as follows:
 - Array name=>value; data type =>char; elements => 15 to accommodate longest string;
2. This activity test the learner's ability to declare an array of characters:
 - `char value[15];`

Activity 14.5: String functions (page 275)

This activity test the learner's ability to identify and use string functions such as `strcat()`, `strlen()` and `strcpy()` listed in the student's book on page 268.

Unit Test 14 (page 275)

1. Array declaration does not initialize an array while initialization give default values
2. For loop is used because the number of elements in an array is known before hand. It is also easier to use.
3. Array name, number of elements, data type
4. It is difficult to detect array overflow because C++ compilers do not include such a facility.

5. Characters occupy one byte while numeric values occupies at least 2 bytes.
6. Null character is used to terminate a string but null value represents nothing
7. `cin.get(arrayname, elements-1)` is a suitable for string input.
8. `strcpy` and `strncpy` are almost similar but `strn` specify number of characters to copied.
10. The program requires the user to declare an array that uses nested if selection, to increment count of each of the six faces of the die declared as
`int faces [6] = {1,2,3,4,5,6};`

11.

```
D:\Rwanda4\C++Programs\ExtraPrograms>  
Hello World!  
12  
p  
Hello World!Computer Science  
Computer Science Study
```

Operating System

Operating System overview

Unit 15

Introduction to Operating Systems

Student's Book page 277 – 302 (18 Periods)

Key unit competence

By the end of this unit, the learner should be able to:

1. Explain the role, types and the Evolution of Computer Operating System.
2. Use Ms-Dos commands.
3. Use Mobile Operating System.

Learning objectives

Knowledge and understanding

- Explain functions, characteristics and components of the operating system.
- List and explain different operating system.
- Identify different type of operating system.
- Describe the use of different MS-DOS and Linux commands.

Skills

- Define operating system Enumerate common computer operating systems.
- Explain different types of operating system.
- Use MS DOS commands.
- Use GUI and commands in Linux.

Attitudes and values

- Appreciate the operating system running in any electronic device.
- Appreciate the touch screen interface of Android on Smartphone and tablets.
- Appreciate the Linux as open source and more resistant against viruses.

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence

supports interpersonal respect and promotes national unity.

Links to other subjects

Mathematics (Logic)

Cross cutting issues addressed in this unit

Assessment criteria

Students can explain the role, the types and the Evolution of Computer Operating Systems, use GUI in Linux, Use commands in Ms-Dos and Linux.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answers
- Discussion
- Role play

Background information

Operating systems introduces the learner on how the resources of the computer are made available to the user. During the course, the student will be able to appreciate the important hardware and software abstraction role that the O/S does to make the applications and user experiences enjoyable.

Suggested teaching/ learning activities

Definition of the operating system

(1 period)

Information to the teacher

Prepare a video clip for the class which

will show the role of the O/S.

Materials: whiteboard, projector, video clip, student text book, .

Preparation

1. Carry with you pictures of the logic role of the O/S.

Additional information teacher

- Guide the students to do learning activity 15.1 and 15.2 pages 277 and 278.

Teaching guidelines 15.1

- After doing activity 15.1 on page 277 let the students go ahead and cover the content in this section. Point the student to Figure 15.1 and 15.2 then discuss with them the structure of O/s.
- Allow the students to make presentations.

Functions of the operating system

(1 periods)

Information to teacher

Materials: student text book, wall charts, whiteboard, projector, movie, computer connected to internet.

Preparation

Come with a picture showing the role of a supervisor in a busy working environment.

Teaching guidelines 15.2

- Divide the students into groups. Let them discuss the functions of the operating system.

Characteristics of O/S

(2 periods)

By the end of this section, the learner should be able to state the characteristics of an O/S.

Materials: A computer connected to the internet, projector, student text book.

Preparation

Carry to class a wall chart showing the characteristics of O/S

Teaching guidelines 15.3

- Start by referring the students to activity 15.3 on page 280.
- After that cover the content in this section.
- Evaluate the learners understanding using formative assessment in class.

Components of the O/S

(2 periods)

By the end of this section, the learner should be able to describe the components of the O/S.

Materials: a computer connected to the internet, student text book, whiteboard, projector.

Teaching guidelines 15.4

- Guide the learners to do activity 15.4 on page 282.
- Cover the content in this section

while giving the students a chance to participate by referring them to Figure 15.2 and 15.3 on page 278 and 282 respectively.

- When it comes to the file system, drive the point home by having the learners browse the file system of the available O/S starting from the root to other files and folders on the computer.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Common Operating systems – UNIX

(2 periods)

By the end of this section, the learner should be able to describe UNIX and use its shell commands.

Materials: a computer connected to the internet, a computer running UNIX, student text book, whiteboard, projector, UNIX manual.

Teaching guidelines 15.5

- Guide learners through the text in the student book. Let them know that UNIX and Linux have the same shell commands so the practice for commands can be done when covering Linux.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Common operating system – LINUX

(1 periods)

By the end of this section, the learner should be able to describe Linux and use GUI and UNIX based shell commands.

Materials: a computer connected to the internet, a computer running Linux, student text book, whiteboard, projector, Linux manual.

Teaching guidelines 15.6

- Guide the learners to do activity 15.6
- Let each student practice how to issue UNIX commands.
- Give each student the Linux manual that you had prepared and let them perform the tasks in it.
- Move round the class to make sure each student is capable of using Linux

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Common operating system – MAC OS X

(1 periods)

By the end of this section, the learner should be able to describe Mac OS X and use its GUI interface.

Materials: a computer connected to

the internet, a computer running Mac OS X, student text book, whiteboard, projector, Mac OS X manual.

Teaching guidelines 15.7

- Guide the learners to search for the Mac OS X desktop pictures on the internet. If you have a demonstration station in class, allow each student a chance to at least start and close common programs on the MAC.
- Let each student practice how to issue Mac OS X commands.
- Give each student the Mac OS X manual that you had prepared and let them perform the tasks in it.
- Move round the class to make sure each student is capable of using Mac OS X

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Common operating system – Windows

(1 periods)

By the end of this section, the learner should be able to describe Windows and use its windows interface.

Materials: a computer connected to the internet, a computer running Windows, student text book, whiteboard, projector, Windows manual.

Teaching guidelines 15.8

- Guide the learners through the text in the student book.

- Let each student practice how to issue Windows commands.
- Give each student the Windows manual that you had prepared and let them perform the tasks in it.
- Move round the class to make sure each student is capable of using Windows.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Common operating system – MS DOS

(2 periods)

By the end of this section, the learner should be able to describe MS DOS and use its GUI interface.

Materials: a computer connected to the internet, a computer running MS DOS, student text book, whiteboard, projector, MS DOS manual.

Teaching guidelines 15.6

- Guide the learners to do activity 15.4 to 15.6 on pages 282 to 286.
- Let each student practice how to issue MS DOS commands.
- Give each student the MS DOS manual that you had prepared and let them perform the tasks in it.
- Move round the class to make sure each student is capable of using MS DOS.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Smartphone O/S Android; Apple (iOS); Windows Phone; BADA; Palm; Blackberry

(6 periods)

By the end of this section, the learner should be able to describe and use various smartphone O/S

Materials: a computer connected to the internet, smart phones or tablets running Android, iOS, Windows, Bada, and Palm.

Teaching guidelines 15.7

- Guide the learners to do activities 15.7 - 15.11 on pages 288 to 292. respectively as you cover each of the above topics per period.
- Let each student practice how to use Android, iOS, Bada, Palm, etc. as is in the student text book.
- teach the content in this section. In case you are not able to finish, give some assignment on the remaining O/S.
- Move round the class to make sure each student is capable of using the smartphone O/S.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

History of the operating system

(2 periods)

By the end of this section, the learner should be able to describe historical development of O/S.

Materials: a computer connected to the internet, a computer student text book, whiteboard, projector, video clip.

Teaching guidelines 15.8

- Guide the learners to do activity 15.12 on page 294.
- Let each student participate in the class activities.
- Cover the content in this section.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Type of operating system

(1 periods)

By the end of this section, the learner should be able to describe types of O/S.

Materials: a computer connected to the internet, wall chart.

Teaching guidelines 15.12

- Guide the learners to do activity 15.13 on page 296.
- Let each student practice participate in the class activities

- Give real life examples where each O/S type is used

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Basic MS-DOS commands and its main features

(1 periods)

By the end of this section, the learner should be able to describe use MS-DOS commands.

Teacher please note you may have to download and install MS-DOS on the computers. The command prompt in Windows may not have all the DOS shell commands in its command.com file.

Materials: a computer connected to the internet, A computer running MS-DOS wall chart.

Teaching guidelines 15.12

- Let each student practice participate in the class activities to cover the content in this section as taught in the student text book.

Evaluate the learners understanding of the concepts presented by going round the class inspecting each of the work done.

Guidelines to Activities

Activity 15.1: Research work (pg. 277)

Brainstorm with the students concerning the scenarios painted in the activity. The idea is to have the learner appreciate the importance of order and how it should be maintained in a situation where different entities are competing for resources. It would be a good idea to visit one of the sites mentioned if possible but it is not a must.

Activity 15.2: Operating systems components (pg. 278)

Guide the learners through this activity with the aim of helping them build on what they already learned from Figure 15.1. Let the learners appreciate how different components work together to achieve the fully functioning O/S

Activity 15.3: Research work (pg. 280)

This activity will build on the research skills of the learner. Identify a good website or some online material and request the student to access and make notes from them. Alternatively, provide relevant materials which they can read and make notes.

Activity 15.4: Research work on operating systems (pg. 282)

This activity will build on the research skills of the learner. Identify a good website or some online material and request the student to access and make

notes from them. Alternatively, provide relevant materials which they can read and make notes.

Activity 15.5: DOS commands (pg. 285)

Guide the student on how to use the basic DOS commands at the DOS prompt. Let the student appreciate the command line interface.

Activity 15.6: DOS commands (pg. 286)

Guide the student on how to use the basic Linux commands at the command prompt. Let the student appreciate the Linux GUI too and how to navigate.

Activity 15.7: Working with smartphone (pg. 288)

Provide at least one smartphone in class. Allow the student to navigate the smartphone starting by the following:

- Logging on to the phone i.e. supplying a password or swiping a pattern to unlock the screen when prompted.
- Accessing Apps e.g. contacts, messages, games, internet etc.

Activity 15.8: Working with android phone (pg. 289)

Provide at least one android phone in class. Allow the student to navigate the smartphone starting by the following:

- Logging on to the phone i.e. supplying a password or swiping a pattern when prompted.
- Accessing Apps e.g. contacts, messages, games, internet etc.

Activity 15.9: Working with apple phone (pg. 290)

Provide at least one apple phone in class. Allow the student to navigate the smartphone starting by the following:

- Logging on to the phone i.e. supplying a password or swiping a pattern when prompted.
- Accessing Apps e.g. contacts, messages, games, internet, camera etc.

Activity 15.10: Working with windows phone (pg. 291)

Provide at least one windows phone in class. Allow the student to navigate the smartphone starting by the following:

- Logging on to the phone i.e. supplying a password or swiping a pattern when prompted.
- Accessing Apps e.g. contacts, messages, games, internet, camera etc.

Activity 15.11: Research on historical development of operating systems (pg. 293)

Let the students continue developing their research skills. This is yet another opportunity for them to traverse the internet and available literature in order to self-learn about the characteristics of the O/S. After the research, you can organize a class discussion then cover the content in this area.

Activity 15.12: Types of operating systems (pg. 294)

Let the students continue developing their research skills. This is yet another opportunity for them to traverse the internet and available literature in order to self-learn about the types of O/S. After the research, you can organize a class discussion then cover the content in this area.

Activity 15.13: How to Learn and Use MS-DOS (pg. 297)

Let the students do this activity to build on what they did earlier in activity 15.5 On page 285. DOS is a command language so although the student may find it difficult at the beginning, they would become more proficient with time as they learn and memorise commands. It would be best to download and install MS-DOS since the command line in Windows may not support some of the more advanced DOS commands.

Answers for Unit 15

Unit Test 15 (page 301)

1. Application software is a user specific program that meets user needs. Operating system is system software that manages the resources of the computer and makes them available to user applications.
3. Shell, User Interface, Commands.
4. Kernel
5. Scheduling, memory management, resource control and management, job scheduling, input/output

- management, job sequencing, security.
6. Usable and interactive, robust, efficient, portable, secure, scalable, extensible
 7. (a) The user types commands on the command line or prompt. The user has to remember all the commands
 (b) Instead of commands the O/S displays menus from which the user selects commands. The user has to be able to identify which command does what.
 (c) GUI: displays commands and programs as icons and menus with a pointer to select them. Very user friendly. Gives tips on what the icons and menus are for therefore easy to learn.
 9. File: storage location of related records on a storage medium; Folder: storage location of related files; Drive: the root storage location that contains the entire file system of a particular storage device; Directory: similar to folder though sometimes it refers to the root folder on a device (drive) that holds all other folders.
 10. (a) Processes data (b) Holds running programs and data (c) help in command / data input/output.
 13. Smartphones are special purpose computers. They have similar resources and are used to perform complex tasks similar computers hence the need for an operating system. They have a fast processor, large memory, many user applications, fast access to the internet etc.
 14. Computer OS are large and complex and perform many tasks; Smartphone OS are lightweight versions of the computer OS and are designed to run on limited hardware resources e.g. to display the user interface on small screens.

Programming

Web technology and Java programming

Unit 16

HTML-Based Web Development

Student's Book page 302 – 342. (20 Periods)

Key unit competence

By the end of this unit, the learner should be able to build standard compliant web pages using HTML Learning objectives.

Learning objectives

Knowledge and understanding

- Explain HTML, XHTML, HTML5 web technologies.
- Differentiate open/empty tag and closed tag in HTML.
- Explain the use of XHTML entities.

Skills

- Create a static website using HTML web technologies by formatting text, images, and page links using HTML tags and their attributes.
- Use appropriately open/empty tags and closed tags.
- Load and format images, audio, and video to a web page.
- Use form controls to design a well organized HTML form.
- Use HTML5 tags and attributes to validate a form using input restrictions.

Attitudes and values

- Design a web page and arrange correctly HTML elements.
- Be able to manage open/empty tags and closed tags.
- Evaluate the use of POST and GET when choosing a correct method to send data.

Generic competences addressed in this unit

Critical thinking: In most of the activities, the student is challenged to apply critical thinking to solve computational problems.

Interpersonal management: In every learning activity that requires group-work or pair work in this unit, the student gets a chance to apply interpersonal and leadership skill.

Co-operation: In every learning activity that requires pairs or group work, the

student gets a chance to nature skill such as co-operation, collaboration and communication.

Science and technology: This unit exposes the learners to the inner working of internet, Web 2.0 and mobile technologies.

Creativity and Innovation: This unit is critical to creativity and innovation because the internet and web helps the student explore various solutions to social and economic challenges.

Links to other subjects

Some of the concepts in this unit are linked to Mathematics and programming languages. This is because the learner is expected to apply mathematical and programming skills in creating dynamic or interactive websites.

Cross-cutting issues addressed in the unit

Though this unit is largely mathematical, we have tried to use the following cross cutting issues:

1. **Gender:** In line with principles of gender equality, we have balance use of male and female across the unit. Where such reference is not required, we have used generic phrases.
2. **Genocide studies:** This unit indirectly addresses this issue through assessment exercises.
3. **Standardization culture:** In this unit

we have addressed standardization culture set by ISO and W3C in HTML4 and HTML5 standards.

Assessment criteria

The students should be able to build standard compliant web pages using HTML. Apart from activities and exercises provided in the student book, the teacher should use other assessment methods and tools to test whether the learner has acquired necessary knowledge and skills required to develop websites.

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion

Background information

Before introducing services available on Internet, it would be a good idea to give a brief overview of how Internet evolved from a project known as ARPAnet conducted by the US department of defense. Make the students also appreciate how internet and the World Wide Web (www) have contributed to current social network, e-commerce and other web services. It would be a good idea also to organize for brainstorming session so that the students identify some of the benefits and limitations of using the internet and World Wide Web.

Suggested teaching/learning activities

The teacher is expected to use demonstrations, research, group discussions and practical exercises to nurture students competence in web development.

Fundamentals of World Wide Web

(1 period)

This topic introduces the learner to basic concepts you may likely come across in the course of accessing internet services.

Information to the teacher

Through demonstration, the teacher should introduce key concepts such as browser, search engine, hypertext, protocols (http, ftp, imap and smtp), web page, website, portal, and meaning of html related terms. You are also required to guide the students on how to get online resources that they can use appreciate importance of World Wide Web. Demonstrate using real life example why www is a spide-like web.

Materials: Computers, Netbeans IDE, Projector and Internet connectivity.

Preparation

This unit may require that computers be connected to internet to demonstrate to the learner some of the activities in the student's book.

Teaching guidelines 16.1

- Organize the class into pairs or groups. Sensitize them on the

importance of working in groups and why each group should have a group leader. Also ensure that each group has a secretary to record and report the group's findings.

- Ask the groups to carry out activities 16.1 and 16.2 on page 303 and 304 respectively. Let the group leader give each member an opportunity to give description and solution to the problem as the secretary notes down the key points.
- The groups should present their solution to the problem in a class discussion through their secretaries.
- Provide a precise summary from their presentation in order to help the learners understand that selection and control structures are instrumental in day-to-day's problem solving.

HTML Syntax and Structure

(3 periods)

This topic introduces the learner to the general syntax of an HTML document structure based on HTML 4.01.

Information to the teacher

The teacher should demonstrate how to create a web page using HTML4 syntax and DTD specification. Once the students have mastered practical exercises, introduce them gradually to creating and manipulating web pages using HTML elements.

Materials: Computers, Dreamweaver, Netbeans IDE, HTML editor, Projector and Internet.

Preparation

This unit may require that computers be connected to internet to demonstrate to the learner some of the concepts in the student's book.

Teaching guidelines 16.2

- Through demonstration, take the student through examples and activities in the student book to help them understand syntax and basic structure of an HTML document.
- Individually or in groups, let the students demonstrate mastery of HTML syntax and structure using a simple web page that displays Hello World on the browser.
- Then evaluate each learner's understanding of the concepts presented before proceeding to the next section that exposes the students to detailed discussion of HTML elements.

HTML Elements

(3 periods)

This topic introduces the learner to the basic HTML4 element based classified into structural, presentational and hypertext elements.

Information to the teacher

The teacher should demonstrate the basic structure of HTML by creating a simple website. It is important to follow Document Object Model (DOM) guidelines provided by W3C when creating HTML documents. The sample

page should serve as the reference for the elements discussed students book.

Materials: Browser, HTML editor, Projector and Internet connectivity.

Preparation

This unit may require that computers be connected to internet to demonstrate to the learner some of the concepts in the student's book.

Teaching guidelines 16.3

- Through demonstration, take the student through examples and activities in the student book to help them understand how to use HTML elements to create web pages.
- Individually or in groups, let the learners work on activities 16.3 on page 307 and 16.4 on page 313 in the student's book.
- It is important that you emphasize on practical work guided demos and remedial sessions to help weak students. You can then evaluate competence before proceeding to the next section.

Introduction to XHTML

(3 periods)

This topic introduces the learner to the general syntax of an XHTML document based on XHTML 1.0 Transitional.

Information to the teacher

The teacher should demonstrate how to create a web page using XHTML 1.0 strict DTD specification. Once

the students have mastered practical exercises, introduce them gradually to creating and manipulating web pages using XHTML. It would be a good idea to let the learner point out benefits and limitations of XHTML and HTML5.

Materials: Computers, Dreamweaver, Netbeans IDE, HTML editor, Projector and Internet connectivity.

Preparation

This unit may require that computers be connected to internet to demonstrate to the learner some of the concepts in the student's book.

Teaching guidelines 16.4

- Through demonstration, take the student through the examples provided in the student's book before they attempt activity 16.5 on page 317. This will help them differentiate between HTML and XHTML in terms of syntax and structure.
- Individually or in groups, let the learner work on assessment exercise 16.1 on page 317 before proceeding to the next section.

Designing HTML Pages

(6 periods)

This topic introduces the learner to design and formatting of web pages using basic HTML elements and attributes. Note that we have deliberately avoided using CSS until the students learn CSS concepts in the next unit.

Information to the teacher

The teacher should demonstrate how to use both HTML4 and XHTML to create a webpage. Once the students have mastered syntax of HTML tags, introduce them gradually to creating websites on their own. It is advisable to let the students work in group to provide solution develop a complete website based on their favourite school activities.

Materials: Computers, Photo editor (e.g. Photoshop), Netbeans IDE, Projector and Internet connectivity.

Preparation

This unit may require that computers be connected to internet to demonstrate to the learner some of the concepts in the student's book.

Teaching guidelines 16.5

- Through demonstration, take the student through examples and activities in the student book to help them understand how to design websites using HTML.
- Individually or in groups, let the learners work on activity 16.6 before exposing them to detailed treatment of this section.
- Step-by-step introduce students on how to insert web contents such as lists, images, hyperlinks, tables and forms. It is advisable to let the students demonstrate their skills by carrying out activities 16.6 to 16.14 from page 317 to 333 of the students' book.

Then evaluate each learner's

understanding of the concepts presented before proceeding to the next section on HTML5.

Introduction to HTML 5

(3 periods)

This topic gives an overview of HTML5 that is expected to be a standard replacement for HTML 5 and XHTML standards.

Information to the teacher

The teacher should demonstrate how to create a web page using HTML5 strict DTD specification. Once the students have mastered practical exercises, introduce them gradually to creating and manipulating web pages using HTML5 and CSS3 as per the W3C recommendations. It would be a good idea to let the learner point out benefits and limitations of HTML5 standard.

Materials: Computers, Dreamweaver, Netbeans IDE, Projector and Internet.

Preparation

Ensure that you have tested and debugged sample programs you intend to use in class to demonstrate various elements of HTML5.

Teaching guidelines 16.6

- Through demonstration, take the student through examples and activities in the student book to help them understand how recursive functions works
- Individually or in groups, let the learners perform activities 16.15

and 16.16 on pages 337 and 340 respectively.

Migrating from HTML4 to HTML5

(1 period)

This topic gives an overview of factors to be considered when migrating from HTML4 to HTML5 in terms of supported elements and browsers. At this point, it would be important to draw the attention of the students to the importance of Cascading Style Sheet (CSS) discussed in the next Unit.

Information to the teacher

The teacher should demonstrate how to convert a website written using HTML4 standard into an HTML5 documents. This is by using HTML5 DOCTYPE, new elements, input types, and constraints. It would be a good idea to let the learner point out benefits and limitations of migration from HTML4 to HTML5 standard.

Materials: Computers, Dreamweaver, Netbeans IDE, Projector and Internet connectivity.

Preparation

Ensure that you have tested and debugged sample HTML documents you intend to use in class to demonstrate how to migrate from HTML4 to HTML5.

Teaching guidelines 16.7

- Through demonstration, take the student through examples and activities in the student book to help

them understand how recursive functions works

- Once the students have mastered the concepts, let them attempt questions on activity 16.17 on page 342 in the student's book.

Answers for Unit 16

Assessment Exercise 16.1 (page 317)

1. Acronyms:
 - (a) HTML - Hypertext Markup Language
 - (b) XHTML - Extensible Hypertext Markup Language
2. HTML tag marks the beginning or end of an element while an element is the component that comprise tag, attribute and content.
3.
 - (a) Title: `<title>content</title>`
 - (b) Body: `<body>content</body>`
 - (c) Paragraph: `<p>content</p>`
 - (d) heading: `<h2>content</h2>`
4. The HTML page follows the DOM that starts with DOCTYPE declaration followed by HTML element within which other elements like head and body are defined.
5. Comments are important ease of understanding of the code by other developers.
6. Using text editor, create the HTML page and save with .html extension
7. Software tools include Dreamweaver, Notepad, Aptana Studio and Notepad++.

Assessment Exercise 16.2 (page 335)

1. Commercial software have more user-friendly features such as templates that simplify design
2. HTML form has controls used to capture data such textboxes, textarea, checkboxes, buttons and lists
3. Widely supported graphical format include GIF (Graphics Interchange Format), JPEG (Joint Photographic Experts Group), and PNG (Portable Network Graphics).
4.
 - (a) Relative URL points to files based on their locations relative to the current file while absolute URL point to files based on their actual locations on the file system. See student book pages 309 and 303.
 - (b) The get method appends data to the URL while post sends data within HTTP body.
 - (c) Tag marks the start or end of element while attribute refers to property of an element.
4. The student book demonstrate how to create and format tables, forms, and images.
5. The get method appends data to the URL while post sends data within HTTP body.

Assessment Exercise 16.3 (page 341)

1.
 - (a) Deprecated elements are HTML features that have been declared obsolete.
 - (b) Pattern is restriction used in reg-

ular expressions.

- (c) Formvalidate: This is to check user input for any mistakes such as entering text instead of numbers.
2. HTML4 and HTML5 Syntax in terms of elements and case sensitivity are almost similar only that HTML has more elements, input types and improved input validation.
3. Factors making it difficult for old browsers to support HTML5 are:
 - (a) Removed for deprecated elements and attributes such as frame set.
 - (b) Use of CSS 3 declaration that are not supported by earlier browsers.
 - (a) New elements and inputs type not compatible with older browsers.
4. By running it on various browserd

Guideline to Activities

Activity 16.1: Evolution of HTML (page 303)

This is an exploratory activity that requires learners to conduct research on evolution of the Internet World Wide Web to current Web 2.0 technologies and services.

Activity 16.2: Evolution of HTML (page 304)

This is an exploratory activity that requires learners to conduct research on how HTML was derived from Standard Generalized Markup Language (SGML) by Tim-Berners Lee.

Activity 16.3: HTML elements and Attributes(page 308)

This is a hands-on activity that introduces the learner to basic HTML4 structure and syntax. The teacher is required to assist the student on proper use of tags and saving with .html extension but not .txt.

Activity 16.4: HTML tags (page 313)

This is a hands-on activity that introduces the learner to basic HTML4 heading and paragraph elements as demonstrated on page 305 and 306 of the student book. Let the learner differentiate between using paragraph tags and leaving a line between paragraphs.

Activity 16.5: XHTML entities and page code (page 316)

1. This is a hands-on activity that introduces the learner to use character entities or codes to display special symbols such as copyright (©/©), greater than (> >) and fractions (½ and ¼). These codes are listed on pages 309 and 310 in the student's book.
2. Let the learners demonstrate how to display the source code, (e.g. on Mozilla, right click web page, then click View Page Source on the context menu)

Activity 16.6: Designing HTML Page (page 317)

This is pre-emptive activity that is intended to prepare the learner for the next section on adding lists, tables,

hyperlinks, form and images. Let the learner search for HTML tutorials from Internet to enhance skills and competence in developing web pages.

Activity 16.7: Ordered List (page 318)

This is a hands-on activity that requires the learner to use the following ordered list type attribute:

```
<ol type = "i"> ...</ol>
```

Activity 16.8: Ordered list (page 320)

This is similar to 16.7 that requires the learner to use the following ordered list type attribute:

```
<ol type = "l"> ...</ol>
```

Activity 16.9: Definition list (page 320)

This is a hands-on activity that requires the learner to use definition list such as demonstrated on pages 306 and 307:

Activity 16.10: Embedding images (page 323)

This is a hands-on activity that requires the learner to use image tag `<img.../>` and its `src` attribute to embed an image. Make the student use both relative and absolute paths to access the image file. For example, the following statements is a relative URL used to access house.png in a folder/directory located one level above the current location of the page (see examples on page 319 of the student's book:

```

```

Activity 16.11: Hyperlinks (page 323)

This is a group-work activity that requires the learners to demonstrate use of hyperlink to direct a visitor to a section of the same page using the following generic syntax:

```
<a href="/html/html_text_links.htm#top">Top</a>
```

Activity 16.12: Tables (page 328)

This is a group-work activity that requires the learners to demonstrate use of table elements and attributes as demonstrated on page 321 of the student book.

Activity 16.13 Form attributes (page 330)

The difference between post and get is well discussed in the student book but the teacher may demonstrate the difference between the two by change the method in the example on page 323 from get to post. The method preferred in sending sensitive data is post.

Activity 16.14: Form controls (page 333)

Use of various form controls are well demonstrated in the example on page 326. However, in this activity, the teacher need to give more practical emphasis on how to insert read-only input, password and radio button elements.

Activity 16.15: HTML5 elements (page 337)

This is a group activity that requires the learners to use Internet to identify and explains new HTML5 elements.

Activity 16.16: HTML5 new input types (page 340)

This is a hands-on activity that requires the learner to use the new HTML5 input called pattern. The acronym regexp stands for regular expression used to restrict form input to a specific values or pattern. The syntax for using regexp in the pattern attribute is:

```
<input pattern="regexp">
```

For example, the HTML5 code below is used to validate telephone number input. Note that the regexp following the pattern has only the plus symbol followed by numeric digits (0-9). The numbers in the curly braces {10,15} means that the number entered must have a minimum of 10 characters and a maximum of 15.

```
<!DOCTYPE html>
<html>
<head>
<title>This</title>
<body>
<form action="demo_form.asp">
Tel: <input type="tel" name="telno"
pattern="[+0-9]{10,15}$">
<input type="submit" value="Send">
</form>
</body>
</html>
```

Activity 16.17: Migrating from HTML4 to HTML5 (page 341)

This is an exploratory activity that requires learners to compare HTML4 and HTML5 and identify some of the challenges of re-engineering legacy web sites comply with requirements of HTML5. This activity may require basic knowledge of CSS discussed in the next unit.

Unit Test 16 (page 342)

1. Web server-Hardware or software used to provide web services on the internet
2. Internet refers to interconnection of computers while web is a service on the internet
3. Web browser
4. HTML4/ HTML5 and XHTML
5. (a) Scrolling image require use of Adobe flash or CSS3
(b) This requires use of element with attribute src pointing to the location of the image
(c) This requires use of <table> element with associated <th>, <tr> and <td>.
5. The statement indicates that user input are to be submitted to the web server by student.php using the get method.
6. The most common controls include: text, textarea, select, radio buttons, checkboxes, file select, command and reset buttons.
8. (a) Hypertext is content accessible on the web by clicking a hyperlink. Hyperlink is a work or phrase you click to move from one web page or section.
(b) XHTML is HTML based on XML while HTML5 relaxes the rule imposed on XHTML but instead introduces new features not compatible with XHTML and HTML4.
9. The relaxed restriction are case sensitive elements, well-formed XML document and doctype declaration.
10. Brower supports, device viewport, (size), HTML version, audience, javascript, and CSS support.

Programming

Web technology and Java programming

Unit 17 Cascading Style Sheets

Student's Book page 343 – 385. (8 Periods)

Key unit competence

By the end of this unit, the learner should be able to build standards compliant web pages using CSS.

Learning objectives

Knowledge and understanding

1. Differentiate html styling and CSS styling.
2. Identify the correct use of a given selector, how to set a selector and how to name different elements to match one CSS selector.
3. Differentiate priorities of styles in a web page namely external CSS, internal CSS and inline CSS.
4. Identify basic properties for different selectors.

Skills

- Create a inline styles to style specific elements inside a web page, internal elements using selectors and external cascading style sheet to differentiate content and its presentation

- Give selectors to html elements(classes, ID) and use html tags to set up their styles.
- Use appropriately these attributes (Font, Margins, Display, Background, Positioning, Floating, Border, Padding) to magnify the presentation of html elements.

Attitudes and values

- Differentiate the content and its presentation using cascading style sheets.

Generic competences addressed in this unit

Problem solving: by doing the learning activities in this unit the student learns how to apply technology to solve problems.

Interpersonal management: working in groups helps nature this skill.

Working together also enhances a sense of respect for each other regardless of socio-cultural differences hence supports interpersonal respect and promotes national unity.

Links to other subjects

HTML programming

Cross cutting issues addressed in this unit

Inclusive education;

Assessment criteria

Students can build standards compliant web pages using HTML and CSS

Suggested teaching methodology

- Guided discovery
- Research
- Question and answer
- Discussion
- Role play

Background information

This is one of the most enjoyable and easiest units in computer science yet students find it very difficult if proper introduction and step by step instruction is not given. Introduce this section with the simple of a web page format using tables. They have already covered this in chapter 16 so they will appreciate the difficulty of making layouts using large tables and the advantage that CSS promises to offer.

Materials: Computers, projector, digital material, text editor (programming in Dreamweaver is better)

Suggested teaching/ learning activities

Definition of CSS

Guide the student to do Activity 17.1 on page 343 as a way of introducing the students to CSS. Since the students have already covered HTML in the previous chapter, they will find it interesting to learn about the new approach to styling web pages.

HTML Styling and disadvantages

The aim of this session is to highlight the disadvantages of HTML and justify the need for CSS by looking at the advantages of CSS.

Guide the students through the content in this section. By the end of this content, the learner should have a very clear understanding of the disadvantages of HTML.

Comparison HTML and CSS

(1 period)

The aim of this session is to introduce the student to CSS. Start from what they know - - HTML. Discuss with them how formatting text using tables is difficult for large web pages.

Information to the teacher

Prepare two web pages, one done by

HTML and the other done by CSS. Let the learners observe both in terms of size, loading time, display capability, beauty to the eye etc.

Materials: whiteboard, projector, student text book.

Preparation

1. Prepare two websites: one formatted with HTML tables, the other with CSS.
2. Let the learner appreciate the need for CSS- demand for mobile web, screens

Additional information teacher

- Guide the students to do learning activity 17.1 on page 343 again.

Teaching guidelines 17.1

- Start by demonstrating two web pages – one formatted by HTML and the other by CSS. Allow the students to observe difference if any e.g. the loading time, appearance etc.
- Proceed to discuss with the students the need for CSS. From the research they did, allow them to present the advantages of CSS, its limitations etc as compared to HTML too.
- Conclude the session by allowing the students to research more about CSS on the internet.

CSS Syntax

(1 periods)

By the end of this section, the learner should be able to demonstrate

understanding of the structure of CSS.

Information to teacher

Materials: Computer running Dreamweaver / or any text editor, student text book, internet connection, web server, browser software.

Preparation

Make sure you download and install a web server on all the computers in the lab (WAMP server is one of the best). If you can get Dreamweaver software, it make it easy for learners to learn how to develop websites using all manner of technologies. Alternatively, a text editor like notepad would do too.

Teaching guidelines 17.2

- Start by giving a small HTML code extract e.g. one for changing the font color to red. Ask the students to tell you what would happen if you have a website of 100 pages and you want to change font color to red.
- Yes, let the student appreciate the amount of repetition, be they tables, styles etc if we use HTML. Now introduce them to the structure of CSS rules (Figure 17.1 on page 345).
- Guide the student to create a small CSS rule so that they can know how to apply a CSS rule.
- Guide the learner to load the CSS page in the server to view how it would appear in a browser e.g. <http://localhost/x.css>.
- Guide the student to do activity 17.2 on page 348.

- Conclude by teaching about the structure of the CSS rule: selector and declarations.

Colors

(1 period)

By the end of this section, the learner should be able to add color to text and CSS pages.

Materials: A computer connected to the internet, text editor, Dreamweaver.

Preparation

Prepare sample code for the Various color formatting strategies e.g. HEX, RGB etc.

Teaching guidelines 17.3

- Provide sample code for color using words like blue, green; RGB and HEX values.
- Guide the student to add colors to CSS web pages using rules.
- Demonstrate to the learner how the code is inserted in the HTML page.

In all the above, let the student test the results in a localhost server.

Adding CSS to Web pages

(2 periods)

By the end of this section, the learner should be able to add CSS to web pages either as inline, internal or external.

Materials: A computer connected to the internet, text editor, Dreamweaver.

Preparation

Prepare sample code for the three CSS insertion strategies:

- External style sheet
- Internal code
- Inline

You will use this to demonstrate to the student how to achieve the required competence.

Teaching guidelines 17.4

- Provide sample code for inline CSS.
- Guide the student to do activity 17.3 and 17.4 pages 351 and 352 respectively.
- Demonstrate to the learner how the code is inserted in the HTML page.
- Demonstrate inline CSS. Let the learner do it practically.
- Demonstrate internal CSS code. Let the student do it practically.
- Demonstrate external CSS code. Let the student do it practically.

In all the above, let the student test the results in a localhost server.

CSS Styles

(4 periods)

By the end of this section, the learner should be able to work with CSS styles like Font, Margins etc.

Materials: A computer connected to the internet, text editor, Dreamweaver.

Preparation

Prepare sample code for the styles::

- Font
- Margins
- Display
- Background
- Positioning
- Floating

You will use this to demonstrate to the student how to achieve the required competence.

Teaching guidelines 17.5

- Provide sample code for all the above styling features at the right time as you cover the relevant content.
- Allow the student to code and run his/her code to see the various styles practically in the browser.
- Guide the student to do Activities 17.5 upto 17.11 on pages 354 to 363 as is relevant during the covering of the content.

In all the above, let the student test the results in a localhost server.

Creating CSS page from Scratch

(4 periods)

By the end of this section, the learner should be able to create a fully functioning CSS page.

Materials: a computer connected to the internet, student text book, whiteboard, projector, text editor, projector.

Teaching guidelines 17.6

- Give the students activity 17.12 and 17.14 on pages 365 and 368 as a projects. Offer facilitation and guidance where necessarily. From time to time, evaluate the students progress and guide the appropriately

Guidelines to Activities

Activity 17.1: Research work (pg. 343)

Guide the learner to do this activity in order for them to self-discover the need and importance of CSS in web development.

Activity 17.2: Research work (pg. 348)

Guide the learner to do this activity. In essence, this is a reminder of what they did in Chapter 16 as you continue setting the platform for CSS programming. This being the first time they are using Style in a HTML page, the learner will obviously be excited.

Activity 17.3: CSS coding strategies (pg. 351)

Guide the learner to do this activity and let them understand when to use which strategy.

Activity 17.4: External CSS example (pg. 352)

Guide the learner to apply external CSS as the example shown in this activity. After coding and running the code correctly challenge the student to come

up with more external CSS examples.

Activity 17.5 – 17.6 Fonts (pg. 354 - 355)

Guide the learner to do this activity. Draw the knowledge of the student using fonts in other applications like MS-Word and let them understand the same concept in the coding here.

Activity 17.7 – 17.8 Margins (pg. 356 - 7)

Guide the learner to do this activity. Draw parallels with margins in word processors to drive the point home.

Activity 17.9 Hiding elements (pg. 358)

Let the students hide and unhide elements practically in CSS.

Activity 17.9 Hiding elements (pg. 358)

Let the students hide and unhide elements practically in CSS.

Activity 17.10 Background example (pg. 360)

Guide the students through this activity. Let each student set a background color using CSS.

Activity 17.11 CSS float property (pg. 363)

Guide the students through this activity.

Let each student set float properties for elements.

Activity 17.12 – 17.13: Setting padding (pg. 365 - 8)

Guide the students through this activity. Let each student set paddings as instructed

Activity 17.14: Creating CSS page example (pg. 368)

Guide the students through this activity. It is a complete CSS development activity.

Activity 17.15: CSS assignment (pg. 384)

Give this assignment to the students. Guide them where they get stuck and require help as they progress.

Answers

Unit Test 17 (pg 385)

- a. style attribute
- b. comma
- c. absolute
- d. :hover
- e. scroll or auto
- f. span, div
- g. -y
- h. clear
- i. text-indent:
- j. margin, border and padding