

**CPT 111**

# Introduction to COMPUTER SCIENCE



**CODeL**

**FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA**  
**CENTRE FOR OPEN DISTANCE AND e-LEARNING**

**FEDERAL UNIVERSITY OF TECHNOLOG, MINNA,  
NIGER STATE, NIGERIA**



**CENTRE FOR OPEN DISTANCE AND  
e-LEARNING (CODEL)**

**B.TECH. COMPUTER SCIENCE  
PROGRAMME**

**COURSE TITLE  
INTRODUCTION TO COMPUTER SCIENCE**

**COURSE CODE  
CPT 111**

**COURSE CODE**

**CPT 111**

**COURSE UNIT**

**2**

**Course Coordinator**

Bashir Mohammed (Ph.D.)  
Department of Computer Science  
Federal University of Technology (FUT) Minna  
Minna, Niger State, Nigeria.

## Course Development Team

---

### CPT 111: Introduction to Computer Science

Subject Matter Experts	Caroline O. Alenoghena (Ph.D.) FUT Minna, Nigeria.
Course Coordinator	Bashir Mohammed (Ph.D.) Department of Computer Science FUT Minna, Niger State, Nigeria.
Instructional System Designers	Oluwole Caleb FALODE (Ph.D.) Bushrah Temitope OJOYE (Mrs.) Centre for Open Distance & e-Learning, FUT Minna, Nigeria
ODL Experts	Amosa Isiaka GAMBARI (Ph.D.) Nicholas Ehikioya ESEZOBOR
Language Editors	Chinenye Priscilla UZOCHUKWU (Mrs.)
Centre Director	Abiodun Musa AIBINU (Ph.D.) Centre for Open Distance & e-Learning FUT Minna, Niger State, Nigeria.

### Introduction

**CPT 111 Introduction to Computer Science** is a 2- credit unit course for students studying towards acquiring a Bachelor of Science in any field. The course is divided into 6 modules and 15 study units. It will first introduce the basic computing terms and the modern history of electronic computers. Next, computer hardware and software are treated in detail. Thereafter, the manner in which data are represented and manipulated in the computer are discussed. This is followed by an extensive discussion of the Von Neumann model of computation. Subsequently, an overview of programming languages is presented. Finally, the student is introduced to the Internet.

The course guide therefore gives you an overview of what CPT 111 is all about, the textbooks and other materials to be referenced, what you expect to know in each unit, and how to work through the course material.

### Recommended Study Time

This course is a 2-credit unit course having 15 study units. You are therefore enjoined to spend at least 2 hours in studying the content of each study unit.

### What You Will Learn in This Course

The overall aim of this course, CPT 111 is to introduce you to basic concepts of computer science. At the end of this course you would have learnt the:

- i. basic computing terminologies
- ii. various computer generations
- iii. various components of computer system both hardware and software
- iv. basics of computer programming languages – types of computer languages, programming paradigms, and
- v. fundamentals of problem-solving using flowcharts and algorithms

### Course Aims

This course aims to introduce students to the basics computing concepts of. It is expected that the knowledge will enable the reader to effectively use computers in his/her profession.

### Course Objectives

It is important to note that each unit has specific objectives. Students should study them carefully before proceeding to subsequent units. Therefore, it may be useful to refer to these objectives in the course of your study of the unit to assess your progress. You should always look at the unit objectives after completing a unit. In this way, you can be sure that you have done what is required of you by the end of the unit.

However, below are overall objectives of this course. On completing this course, you should be able to:

- (i). Define basic computing terms
- (ii). Explain the computer generations
- (iii). Describe the components of a computer system

- (iv). Describe the categories of computer hardware
- (v). Discuss the types of computer software
- (vi). Explain the types and functions of operating systems
- (vii). Describe how data are stored and manipulated in a computer
- (viii). Discuss fundamental concepts of digital logic
- (ix). Explain the Von Neumann model of computation
- (x). Mention the types of programming languages
- (xi). Differentiate between compilers and interpreters
- (xii). Describe procedural and object-oriented programming paradigms
- (xiii). Write algorithms to carry out simple tasks
- (xiv). Draw flowcharts to solve simple problems
- (xv). Describe Internet-related concepts such as HTML protocol and the format of webpages

### Working Through This Course

In order to have a thorough understanding of the course units, you will need to read and understand the contents, practice the steps by designing and implementing a mini system for your department, and be committed to learning and implementing your knowledge.

This course is designed to cover approximately sixteen weeks, and it will require your devoted attention. You should do the exercises in the Tutor-Marked Assignments and submit to your tutors.

### Course Materials

The major components of the course are:

1. Course Guide
2. Study Units
3. Text Books
4. Assignment File
5. Presentation Schedule

### Study Units

There are Fifteen (15) Study Units and Eight (8) Modules in this course. They are:

<b>Module One</b>	<b>Introduction to Computing</b>	
	Unit 1	Basic Concepts
	Unit 2	Computer Hardware
<b>Module Two</b>	<b>Computer Software</b>	
	Unit 1	Introduction to Computer Software
	Unit 2	Operating Systems
<b>Module Three</b>	<b>Machine Level Representation of Data</b>	
	Unit 1	Units of Data and Fundamental Operations on Bits

	Unit 2	Number Bases and Types
	Unit 3	Representation of Non-Numeric Data, Records and Registers
<b>Module Four</b>	<b>Logic</b>	
	Unit 1	Digital Logic
	Unit 2	Memory
	Unit 3	Von Neumann Model of Computation
<b>Module Five</b>	<b>Programming the Computer</b>	
	Unit 1	Overview of Programming Languages
	Unit 2	Algorithms
	Unit 3	Flowcharts
<b>Module Six</b>	<b>The Internet</b>	
	Unit 1	Introduction to the Internet
	Unit 2	Web Technology

### Recommended Texts

The following texts and Internet resource links will be of enormous benefit to you in learning this course:

Bonk, C. J., (2009). *The world is open: How web technology is revolutionizing education*. John Wiley & Sons

Brookshear J. G. (2005), *Computer Science: An Overview*, 9th Edition, Pearson Addison Wesley.

Hodgkins, R. (n.d.). Understanding web technology. Retrieved June 3, 2015, from <http://www.computerweekly.com/feature/Understanding-web-technology>

Kurose, Ross (2006), *Computer Networking, A Top-Down Approach* 6<sup>th</sup> Edition

Leon, A., Leon, M., (1999), *Fundamentals of Information Technology*, Leon Vikas Hall

Levitin, A., (2007), *Introduction to the Design Analysis of Algorithms:2nd Edition*, Pearson Addison Wesley

Norton, P. (2005), *Peter Norton's Introduction to Computers* 6th edition, McGraw-Hill/Irwin

Null, L., Labur, J., (2003), *The Essentials of Computer Organization and Architecture*.

Parsons, J. J., Oja, D. (2011), *Practical Computer Literacy* 3rd edition, Centage Learning.

Pfaffenberger, B. (2002), *Computers in your Future* 4th edition, Prentice Hall

Rajaraman V and Radhakrishanan T, (2009), *An Introduction to Digital Computer Design*, 5th Edition, PHI Learning Private Limited

Silberschatz, A., Galvin, P. B., Gagne, G., (2010), *Operating Systems Concepts*, 8<sup>th</sup> ed, John Wiley & Sons, New Jersey

Stallings, W., (2012), *Operating systems: Internals and design principles*, 7<sup>th</sup> ed, Prentice Hall, New Jersey

Tanenbaum, A., (2009), *Modern Operating Systems*, 3<sup>rd</sup> ed, Prentice Hall, New Jersey

## Assignment File

The assignment file will be given to you in due course. In this file, you will find all the details of the work you must submit to your tutor for marking. The marks you obtain for these assignments will count towards the final mark for the course. Altogether, there are tutor marked assignments for this course.

## Presentation Schedule

The presentation schedule included in this course guide provides you with important dates for completion of each tutor marked assignment. You should therefore endeavor to meet the deadlines.

## Assessment

There are two aspects to the assessment of this course. First, there are tutor marked assignments; and second, the written examination. Therefore, you are expected to take note of the facts, information and problem solving gathered during the course. The tutor marked assignments must be submitted to your tutor for formal assessment, in accordance to the deadline given. The work submitted will count for 40% of your total course mark.

At the end of the course, you will need to sit for a final written examination. This examination will account for 60% of your total score. You will be required to submit some assignments by uploading them to CPT 111 page on the u-learn portal.

## Tutor Marked Assignments (TMAs)

There are TMAs in this course. You need to submit all the TMAs. The best 10 will therefore be counted. When you have completed each assignment, send them to your tutor as soon as possible and make certain that it gets to your tutor on or before the stipulated deadline. If for any reason you cannot complete your assignment on time, contact your tutor before the assignment is due to discuss the possibility of extension. Extension will not be granted after the deadline, unless on extraordinary cases.

## Final Examination and Grading

The final examination for CPT 111 will last for a period of 2 hours and has a value of 60% of the total course grade. The examination will consist of questions which reflect the self-assessment questions and tutor marked assignments that you have previously encountered. Furthermore, all areas of the course will be examined. It would be better to use the time between finishing the last unit and sitting for the examination, to revise the entire course. You might find it useful to review your TMAs and comment on them before the examination. The final examination covers information from all parts of the course.

## The following are Practical Strategies for Working through this Course

1. Read the course guide thoroughly
2. Organize a study schedule. Refer to the course overview for more details. Note the time you are expected to spend on each unit and how the assignment relates to the units. Important details, e.g. details of your tutorials and the date of the first day of the semester are available. You need to gather together all this information in one place such as a diary,



a wall chart calendar or an organizer. Whatever method you choose, you should decide on and write in your own dates for working on each unit.

3. Once you have created your own study schedule, do everything you can to stick to it. The major reason that students fail is that they get behind with their course works. If you get into difficulties with your schedule, please let your tutor know before it is too late for help.
4. Turn to Unit 1 and read the introduction and the objectives for the unit.
5. Assemble the study materials. Information about what you need for a unit is given in the table of content at the beginning of each unit. You will almost always need both the study unit you are working on and one of the materials recommended for further readings, on your desk at the same time.
6. Work through the unit, the content of the unit itself has been arranged to provide a sequence for you to follow. As you work through the unit, you will be encouraged to read from your set books
7. Keep in mind that you will learn a lot by doing all your assignments carefully. They have been designed to help you meet the objectives of the course and will help you pass the examination.
8. Review the objectives of each study unit to confirm that you have achieved them.
  - a. If you are not certain about any of the objectives, review the study material and consult your tutor.
9. When you are confident that you have achieved a unit's objectives, you can start on the next unit. Proceed unit by unit through the course and try to pace your study so that you can keep yourself on schedule.
10. When you have submitted an assignment to your tutor for marking, do not wait for its return before starting on the next unit. Keep to your schedule. When the assignment is returned, pay particular attention to your tutor's comments, both on the tutor marked assignment form and also written on the assignment. Consult you tutor as soon as possible if you have any questions or problems.
11. After completing the last unit, review the course and prepare yourself for the final examination. Check that you have achieved the unit objectives (listed at the beginning of each unit) and the course objectives (listed in this course guide).

## Tutors and Tutorials

There are few hours of tutorial provided in support of this course. You will be notified of the dates, time and location together with the name and phone number of your tutor as soon as you are allocated a tutorial group. Your tutor will mark and comment on your assignments, keep a close watch on your progress and on any difficulties you might encounter and provide assistance to you during the course. You must mail your tutor marked assignment to your tutor well before the due date. At least two working days are required for this purpose. They will be marked by your tutor and returned to you as soon as possible.

Do not hesitate to contact your tutor by telephone, e-mail or discussion board if you need help. The following might be circumstances in which you would find help necessary: contact your tutor if:

1. You do not understand any part of the study units or the assigned readings.
2. You have difficulty with the self test or exercise.
3. You have questions or problems with an assignment, with your tutor's comments on an assignment or with the grading of an assignment.

You should endeavour to attend the tutorials. This is the only opportunity to have face to face contact with your tutor and ask questions which are answered instantly. You can raise any problem encountered in the course of your study. To gain the maximum benefit from the course tutorials, have some questions handy before attending them. You will learn a lot from participating actively in discussions.

GOOD LUCK!

## Table of Content

---

<b>Course Development Team.....</b>	<b>iii</b>
<b>Study Guide.....</b>	<b>iv</b>
<b>Table of Content.....</b>	<b>ix</b>
<b>Module 1: Introduction to Computer.....</b>	<b>1</b>
Unit 1: Basic Concepts.....	2
Unit 2: Computer Hardware.....	8
<b>Module 2: Computer Software.....</b>	<b>14</b>
Unit 1: Introduction to Computer Software.....	15
Unit 2: Operating Systems.....	21
<b>Module 3: Machine Level Representation of Data.....</b>	<b>27</b>
Unit 1: Units of Data and Fundamental Operation of Bits.....	28
Unit 2: Number Bases and Types.....	37
Unit 3: Representation of Non Numeric Data, Record and Registers.....	47
<b>Module 4: Logic.....</b>	<b>52</b>
Unit 1: Digital Logic.....	53
Unit 2: Memory.....	60
Unit 3: Von Neumann Model of Computation.....	67
<b>Module 5: Programming of Computer.....</b>	<b>73</b>
Unit 1: Overview of Programming Languages.....	74
Unit 2: Algorithms.....	84
Unit 3: Flowcharts.....	90
<b>Module 6: The Internet.....</b>	<b>97</b>
Unit 1: Introduction to the Internet.....	98
Unit 2: Web Technology.....	105

# Module 1

---

## Introduction to Computing

Unit 1:           Basic Concepts  
Unit 2:           Computer Hardware

# Unit 1

---

## Basic Concepts

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Definition of Terms
  - 3.2 Computer Generations
  - 3.3 Components of a Computer System
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Reading

## 1.0 Introduction

---

You would agree recent time; computers have become very important tools in virtually all aspects of human lives. The development of handheld computers such as personal digital assistants (PDAs), tablet computers and smart phones has helped computers to become known. We will be discussing present background information about computers by defining basic computer terms, providing a brief history of modern computing, and describing the parts of a computer system unit.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to:

- i. Define basic terms such as computer, data and information
- ii. Explain the different generations of computers
- iii. Describe the constituents of a computer system

## 3.0 Learning Contents

---

### 3.1 Definition of Terms

---

**Computer:** A computer is an electronic device that processes data, in order to convert it to information that is useful to people. A computer can perform the following tasks:

- i. Accept data through an input device (e.g. keyboard or mouse)
- ii. Process the data to convert it into information
- iii. Display the information on an output device (e.g. visual display unit or printer)
- iv. Store the information for future use in a storage device (e.g. hard disc or compact disc)

**Data:** Data refer to raw or unprocessed facts about a person, place or thing. Examples of data include name, age, height and profession. Data is the plural for datum.

**Information:** Information is processed data or data that has been converted into useful form e.g. the result of students in an examination or the net pay of an employee.

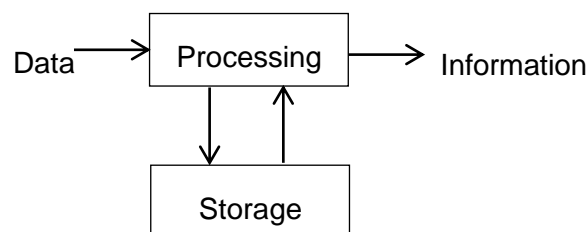


Figure 1.0 Schematic diagram of a compute

## Self-Assessment Exercise(s) 1

1. Define a computer
2. Information is processed data (True/False)
3. Data is processed information (True/False)
4. List four (4) tasks that a computer can perform

### 3.2 Computer Generations

---

You would agree with me that computer scientists and historians use the term *computer generations* to describe the stage-by-stage development of modern computing. Each generation is characterized by the technology used to fabricate computers at that time.

#### 3.2.1 The First Generation (1950s)

In 1951, Presper Eckert and John Mauchly delivered the Universal Automatic Computer (UNIVAC), the first successful general-purpose computer to the U.S. Census Bureau. The following year, the UNIVAC gained fame when it correctly predicted that Dwight Eisenhower will win the U.S. presidential election. The UNIVAC used punched cards and magnetic tape for input. It was made of vacuum tubes, which required lots of power and failed regularly. The UNIVAC was programmed using machine language, which is composed of strings of zeros and ones.

#### 3.2.2 The Second Generation (1960s)

I want you to understand that First-generation computers were highly unreliable because the vacuum tubes burned out frequently. Second generation computers were made of transistors, which are small electronic devices that can control the flow of electricity in an electronic circuit. Owing to the use of transistors, second generation computers were faster, smaller, and more reliable than first-generation computers. Even though second generation computers still used punched cards for input, they had printers, tape storage, and disk storage.

Do you know that Second generation computers were programmed using high level programming languages, rather than machine language? Unlike machine language which was cumbersome to work with, high-level programming languages are much easier for people to understand and work with because of the use of English commands and mathematical symbols. So also, unlike machine language which was machine-dependent, a high-level language program can be used on computers produced by different manufacturers. An example of second generation computers is the IBM's line of computers called System/360.

#### 3.2.3 The Third Generation (Mid-1960s to Mid-1970s)

Be informed that the key distinction of third generation computers was their use of Integrated Circuits (ICs). ICs incorporated many transistors and electronic circuits on a single silicon chip. They were much cheaper than transistors. Using a technology called small-scale integration (SSI), the earliest ICs could contain 10 to 20 transistors on a chip. By the late 1960s, medium-scale integration (MSI) allowed between 20 and 200 transistors to be placed on a chip. In the early 1970s, large-scale integration (LSI) was achieved, allowing up to 5,000 transistors on a single chip. ICs made it possible to produce smaller, inexpensive computers that more organizations could afford to buy.

There was another innovation of third generation computers with timesharing. A technique that allowed several people to simultaneously use a computer from their remote terminals. An example of third generation computers is DEC's PDP-8.

### 3.2.4 The Fourth Generation (1975 to the Present)

Have it in mind that the development of increasingly complex ICs gave rise to very-large-scale integration (VLSI) technology, which allowed the entire processing circuits of a computer to be placed on a single chip. Dr. Ted Hoff built the Intel 4004, the world's first microprocessor, which had 2,300 transistors. A microprocessor is a single chip that holds the entire control unit and arithmetic-logic unit of a computer.

Be informed that shortly after the introduction of the 4004 and its successor the 8008, the first microcomputers began to appear. A microcomputer is a computer that uses a microprocessor as its central processing unit. In other words, the microprocessor is responsible for processing data in a microcomputer. Apple Computer, Inc. developed the Apple I microcomputer, followed by the highly successful Apple II. The Apple II was based on the Motorola 6502 microprocessor and it had a keyboard, monitor and floppy disk drive. The development of VisiCalc, the first electronic spreadsheet software in 1979 showed that the Apple II was not just a toy to be used in schools and homes; it meant that microcomputers could be used for business applications. In 1981, the IBM Personal Computer (PC), was released. The PC used the Intel 8080 microprocessor as well as Microsoft's operating system called MS-DOS.

The first microcomputers were not easy to use because users had to type commands on the command line to perform such actions as formatting a disk or starting a program. However, from the mid-80s, graphical user interfaces (GUIs) were incorporated into microcomputers, allowing users to interact with on-screen, icons, windows and pull-down menus using a pointing device such as a mouse. GUIs were easier to use than command line interfaces because they eliminated the need to memorize commands.

#### Self-Assessment Exercise(s) 2

1. First generation computers were made of .....
2. A transistor is larger than a vacuum tube (True/False).
3. Machine language is more cumbersome to work with than high level programming languages (True/False).
4. Third generation computers were made of .....
5. .... is a single chip that holds the entire control unit and arithmetic-logic unit of a computer.

### 3.3 Components of a Computer System

---

Every computer is part of a system. The complete computer system consists of four parts, namely: hardware, software, data and user.

#### Hardware

Hardware are the electrical and mechanical devices that make up a computer. They are the parts of the computer that can be touched and felt, such as keyboard, mouse, monitor, speaker and printer.



## Software

Software is the set of instructions that tells the computer what task to do, and how to do it. A piece of software is referred to as a program. Some programs such as operating systems exist to help the computer perform tasks and manage its resources. Other programs such as word processors and spreadsheets allow users to create and manage documents.

## Data

Data refers to individual facts that may not make sense on their own. The computer's job is to convert data into useful information

## Users

The people who operate computers systems are referred to as users. Even though a computer may function without anyone sitting in front of it and operating it, no computer is completely autonomous. Human beings are still needed to design, build, program and repair computers.

## 4.0 Conclusion

---

We have learnt so far that computer is a device that transforms raw data into useful information. A complete computer system is made up of the physical components (hardware), instructions (software), operators (users) and data.

## 5.0 Summary

---

You have learnt that:

- i. Data are raw facts, while information refers to processed data
- ii. Computer generations are used to classify the history of modern computers according to their key characteristics.
- iii. The complete computer system is made up of four parts, namely: hardware, software, data and user

## 6.0 Tutor-Marked Assignment

---

1. What is a computer?
2. Explain the difference between data and information.
3. Mention four tasks that a computer can perform.
4. What major hardware technology characterized each of the four generations of computers?
5. What are the differences between a command line interface and a graphical user interface? Which one is easier to use and why?
6. Describe the four components of a computer system

## 7.0 References/ Further Readings

---

Norton, P. (2005), *Peter Norton's Introduction to Computers* 6th edition, McGraw-Hill/Irwin  
Pfaffenberger, B. (2002), *Computers in your Future* 4th edition, Prentice Hall

# Unit 2

---

## Computer Hardware

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Component of Computer Hardware
  - 3.2 Central Processing Unit
  - 3.3 Memory
  - 3.4 Storage Devices
  - 3.4 Input Devices
  - 3.5 Output Devices
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Reading

## 1.0 Introduction

---

Computer hardware refers to the mechanical and electronic parts of a computer that can be touched and seen. Hardware is what the layman often refers to as the computer. This unit discusses the different parts of computer hardware.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to:

- i. List the main parts of a computer hardware
- ii. Describe the different hardware components

## 3.0 Learning Contents

---

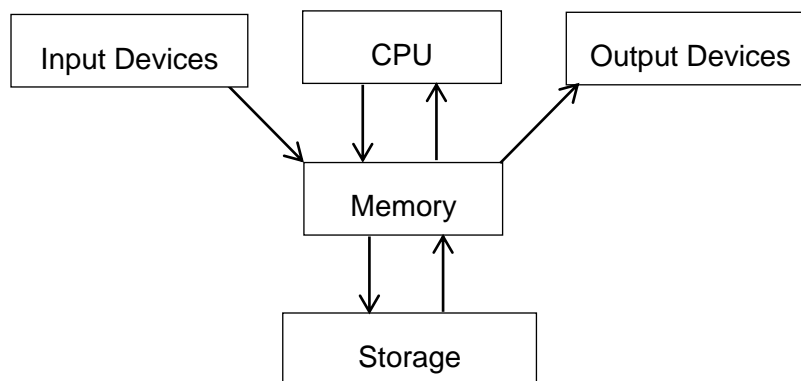
### 3.1 Components of Computer Hardware

---

The following hardware components are required by all general-purpose computers:

- i. Central processing unit (CPU) for executing instructions
- ii. Memory for storing data and programs, at least temporarily
- iii. Input devices for sending data and instructions into the computer
- iv. Output devices for bringing retrieving information from the computer
- v. Storage devices for retaining large amounts of information permanently.

Figure 2.0: Shows how the different hardware components interact in a general-purpose computer.



**Figure 2.0** Parts of the Computer Hardware

### 3.2 Central Processing Unit

---

The central processing unit or simply the processor, is the part of the computer that executes program instructions and controls the operation of all other hardware components. It is sometimes described as the computer's brain. In a microcomputer, the CPU is contained in a single chip and referred to as a microprocessor.

The CPU is made up of three parts- the arithmetic-logic unit, control unit and registers.

**Control Unit:** It coordinates other hardware components in order to carry out program instructions. During program execution, it directs electronic signals between memory and arithmetic-logic unit as well as between CPU and input/output devices.

**Arithmetic-Logic Unit:** The arithmetic-logic unit (ALU) performs two types of operations – arithmetic operations and logical operations. Arithmetic operations are fundamental mathematical operations such as addition, subtraction, multiplication and division. Logical operations compare two values to determine whether one is less than, greater than or equal to the other.

**Registers:** Registers are high speed storage locations within the CPU that temporarily hold data and instructions during processing. Because they are part of the CPU, register contents can be handled much faster than the contents of memory.

### Self-Assessment Exercise(s) 1

1. List the main components of computer hardware.
2. Which of the following store data permanently? a) registers b) memory c) storage
3. Contents of memory are processed faster than those of registers (True/False).
4. Which of the following is not a part of the CPU? a) memory b) registers c) ALU d) control
5. Unit

## 3.3 Memory

---

General-purpose computers have two kinds of memory:

- i. Random Access Memory (RAM) and
- ii. Read Only Memory (ROM).

**Random Access Memory:** RAM is the working memory of the temporary area for holding raw data for processing, instructions for processing the data, and information. It is the working memory of the computer. When a program is started, it is loaded from storage to RAM. The program remains in RAM until it is closed. When data is entered into the computer through the input device, it is first stored in RAM. During processing, data and instructions are fetched from RAM and stored in CPU registers. At the end of processing, results are stored in RAM before they can be displayed to the user.

There are two reasons why RAM is a temporary storage:

- i. RAM is volatile, meaning that it requires electric power to hold data. When the computer is turned off, everything stored in RAM disappears.
- ii. Data stored in RAM can easily be changed

**Read Only Memory:** ROM is a type of memory that holds the built-in instructions that tell the computer what to do when it is turned on. Unlike RAM which is temporary and volatile, ROM is permanent and non-volatile. Instructions stored in ROM cannot be changed, and they are preserved even when the computer is turned off.

### 3.3 Storage Devices

---

Do you know that storage devices are non-volatile, long-term memories Unlike Ram whose contents are lost when power supply goes off, storage devices preserve their contents even when there is no power supply? As a result, storage devices are commonly used to transfer data and programs from one computer to another. They can also be used to back up valuable data, so that the data can be restored after a computer crash results in loss of data. In comparison to ROM whose contents cannot be changed, information held in storage devices can be easily replaced. Examples of storage devices are hard disks, optical discs and flash drives.

#### Hard Disks

A hard disk is the main storage device in most computers. It is slower, less expensive and has higher capacity than RAM. A hard disk is made up of several platters, which are coated with magnetic material. Each platter requires read/write heads to retrieve or modify the information stored in the platter.

#### Optical Disks

Optical disks allow data to be read (i.e., retrieved) or written (i.e., changed) by lasers. The process of creating an optical disk is called burning. Even though optical disks can tolerate temperature fluctuations better than hard disks, they can easily get scratched hence they need to be stored in a case or jacket. The types of optical disks include:

- i. Compact Disk Read Only Memory: Compact Disk Read Only Memory (CD ROM) contains data stored on the disk while it was being manufactured. The data can be read many times but cannot be modified
- ii. Compact Disk Write Once Read Many times: Data can be written into a Compact Disk Write Once Read Many times (CD WORM) once, but after that it behaves like a CD ROM.
- iii. Compact Disk Rewriteable: A Compact Disk Rewriteable (CD RW) allows data to be written and read as many times as desired, just like a hard disk.

**Universal Serial Bus Flash drives:** Universal Serial Bus Flash drives or simply flash drives are amongst the most popular storage devices. There are several reasons for their popularity:

- i. They are portable i.e., easy to carry about
- ii. Virtually all computers have USB ports into which flash drives can be plugged
- iii. Flash drives can be used without bulky devices or burners

#### Self-Assessment Exercise(s) 2

1. State one similarity and one difference between RAM and Storage.
2. State one similarity and one difference between ROM and Storage.
3. Give reasons why flash drives are very popular

### 3.3 Input Devices

---

Are you aware that an input device is any machine used to send data and instructions into the computer? Examples of input devices are keyboard, mouse, joystick, digital cameras and scanners

**Keyboard:** The keyboard is the most important input device on a computer. Computer keyboards have the same layout as standard typewriters. The layout is called QWERTY, because the first six letters on the top row of letters are Q, W, E, R, T and Y. In addition to typing keys, computer keyboards have arrow keys for navigation, function keys for computer-specific tasks, and a calculator-like numeric keypad.

**Mouse:** A mouse is a device for controlling the movement of a pointer or cursor on the screen. It consists of a ball which is rolled on a surface. The mouse derives its name from its resemblance to a real mouse.

**Joystick:** A joystick resembles a car's gear stick. Moving the stick in any direction results in a corresponding movement of an on-screen object such as a pointer. Joysticks are mostly used for playing computer games.

**Digital cameras:** They are used to capture still or moving images, which can be transferred to a computer and manipulated using image or video processing software.

### 3.4 Output Devices

---

Let us talk about output devices, it retrieves information from the computer and presents it to the user. Examples of output devices are monitors, printers, plotters and speakers.

**Monitors:** The monitor is the computer's display screen. It is also known as visual display unit. The two commonly used technologies for monitors are liquid crystal display and light emitting diodes, which have replaced cathode ray tube monitors that resembled bulky old-fashioned televisions.

**Printers:** A printer is a device that expresses text or illustration on paper and other media. The two most commonly used types of printers today are inkjet and laser printers. Inkjet printers work by spraying ink on papers to produce text and characters. On the other hand, laser printers pass a beam on a cylindrical drum, which collects powdered ink (toner) and transfers the toner to paper. Laser printers work in the same way as photocopiers.

**Plotters:** A plotter is a device that draws pictures by moving one or more pens on paper. Plotters can be used to produce large printouts. They are typically used by engineers, architects and interior designers who require more precision than can be offered by printers.

## 4.0 Conclusion

---

We should understand that the computer is a device that transforms raw data into useful information. A complete computer system is made up of the physical components (hardware), instructions (software), operators (users) and data.

## 5.0 Summary

---

This unit has described basic computer terms and discussed the history of modern computers. Furthermore, the various components of a computer system have been described in the unit.

## 6.0 Tutor-Marked Assignment

---

1. What is a CPU? Describe the components of a CPU.
2. Describe the following terms:
  - i. RAM
  - ii. ROM
3. What is an output device? Briefly describe any three output devices.
4. What is an input device? Briefly describe any three input devices.

## 7.0 References/ Further Readings

---

Parsons, J. J., Oja, D. (2011), *Practical Computer Literacy* 3rd edition, Centage Learning.  
Norton, P. (2005), *Peter Norton's Introduction to Computers* 6th edition, McGraw-Hill/Irwin  
Leon, A., Leon, M., (1999), *Fundamentals of Information Technology*, leonVikas Hall

# Module 2

---

## Computer Software

Unit 1: Introduction to Computer Software

Unit 2: Operation Systems



# Unit 1

---

## Introduction to Computer Software

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Categories of Computer Software
  - 3.2 Operating Systems
  - 3.3 Utilities
  - 3.4 Languages
  - 3.5 Word Processors
  - 3.6 Spreadsheets
  - 3.7 Database Management Systems
  - 3.8 Presentation Software
  - 3.9 Graphics Software
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

You should be familiar with hardware which refers to the physical parts of the computer that can be touched, software is the set of intangible instructions that tell the computer what to do. This unit discusses the two major categories of computer software, namely application and system software, as well as the types of software in each category.

## 2.0 Learning outcomes

---

At the end of studying this unit, you should be able to:

- i. Describe the two broad categories of computer software
- ii. Describe different types of software such as operating systems, utilities and language translators.

## 3.0 Learning Contents

---

### 3.1 Categories of Computer Software

---

Be informed that software is broadly divided into two categories:

**System Software:** These are the software that interact with the computer at a very basic level. They help the computer to carry out its basic operating functions such as managing files, interacting with input/output devices, and removing viruses from the computer. Examples of system software are operating systems, utilities and language translators.

**Application Software:** These are the software that help users to do real work such as creating documents, editing photos or tracking finances. Application software do not interact with the computer directly. Rather, they do so through system software such as operating systems and utilities.

Figure 1.0, provides an overview of application software and system software, as well as the types of software in each category.

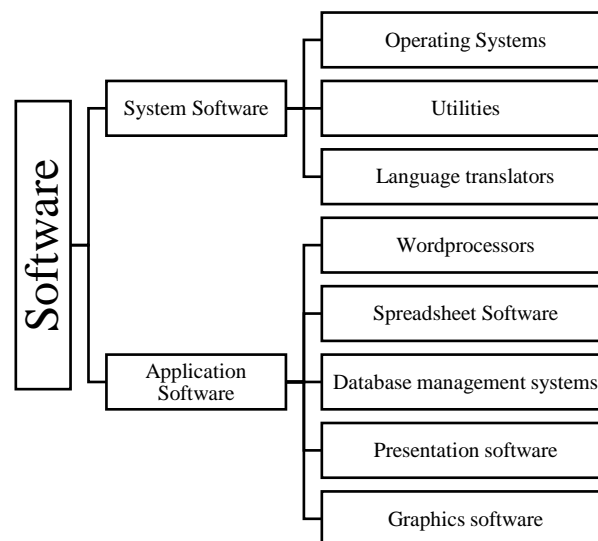


Figure 1.0: Overview of types of software

## 3.2 Operating Systems

---

Operating systems are the most important programs that run on a computer, because they control all activities that take place in a computer. Operating systems perform basic tasks such as keeping track of files, management of memory allocated to programs and data, and controlling input/output devices such as keyboard and printer.

Examples of operating systems for general-purpose computers are Microsoft Windows, Mac OS, UNIX and Linux. Handheld devices use operating systems such as Windows Mobile OS, iOS and Android OS.

## 3.3 Utilities

---

A utility is a program designed to perform tasks such as making a computer's performance efficient, protecting data and make communication easy. Examples of utility programs are:

- i. **Antivirus software** for protecting the computer from computer virus.
- ii. **Backup software** that help store copies of files, which can be restored in case the files get lost or damaged.
- iii. **Compression utilities (or zip software)** that shrink files so that they require less storage space. Compression utilities can also return files to their original form when required.
- iv. **Email software** for managing the flow of data to/from an electronic post office box.

## 3.4 Language Translators

---

Now know that computers understand only machine language, in which instructions are written as strings of zeros and ones. However, programming in machine language is difficult and error-prone for humans. As a result, programs are usually written in high level languages which contain English commands and mathematical symbols that humans are familiar with.

Language translators are system software that convert high level language programs to machine language programs which the computer understands. Examples of language translators are:

- i. **Compilers** which translate all instructions in a high level language program to machine language before executing any instruction.
- ii. **Interpreters** which translate and execute an instruction in a high level language program before moving to the next instruction.

### Self-Assessment Exercise(s) 1

1. ....software helps the computer to carry out its basic operating functions.
2. ....software helps to do real work.
3. Which of the following is NOT a type of *utility* program?
  - a. Antivirus software
  - b. backup software
  - c. Compilers
  - d. Compression or zip software
4. Which of the following is NOT an example of *language translators*?
  - a. Interpreters

- b. Debuggers
- c. Compilers

### 3.5 Word Processors

---

A word processor is a program for producing documents such as letters, memos, reports and manuscripts. Word-processing software allow you to create, edit, format and spell-check documents on the screen before orienting on paper. Examples of word processors are Microsoft Word and OpenOffice Writer.

### 3.6 Spreadsheets

---

You need to know that spreadsheet is an arrangement of rows and columns containing values that can be manipulated. An electronic spreadsheet is a program for working with spreadsheets. Because electronic spreadsheets allow users to simple or complex formula, they can be used for performing tasks such as budgeting, tracking finances, calculating loan payments, computing student grades and estimating project costs. Electronic spreadsheets allow users to display varieties of colorful graphs. They often support *what-if-analysis*, for analyzing different scenarios such as “What if I score Bs in all my courses this semester? But what if I score only Cs?” examples of electronic spreadsheets are Microsoft Excel and OpenOffice Calc.

### 3.7 Database Management Systems

---

However, database is a collection of data stored on one or more computers. A database can contain data such as details of books in a library, university student records, or bank customer details. A database management system (DBMS) is a program for storing, modifying, finding and reporting data contained in a database. DBMS allow users to *query the database*, that is, to extract information that meets certain criteria. For example, a query could request the DBMS to list all students who scored at least a B grade in CPT 111, arranged according to their matriculation numbers. Examples of popular DBMS are Microsoft Access, Oracle and My SQL.

### 3.8 Presentation Software

---

Be aware that Presentation software enable users to combine text, graphs, photos, sound clips and animation into series of electronic slides. For one-on-one presentations, the slides can be viewed on the monitor. However, group presentations are commonly viewed using a computer projector. Presentation slides can also be posted on the Internet. Electronic slides can be used by instructors and students to deliver lecturers and oral presentations. They are also useful for presenting ideas in meetings and conferences. Examples of common presentation software are Microsoft PowerPoint and OpenOffice Impress and Google Presentations.

### 3.9 Graphics Software

---

Graphics software allow users to create, edit and manipulate graphics. These graphics could be pictures, images, drawings, icons or photographs. There are different types of graphics software:

- i. **Paint Software** help you paint images by providing pens, brushes and paints. They represent images using bitmap graphics formats such as BMP, PNG, TIF and JPEG. Examples of paint software are Microsoft Paint and Corel Painter.
- ii. **Drawing Software** are designed to for combining lines, shapes and colors into diagrams. They represent diagrams using vector graphics files such as WMF and EPS. Examples of drawing software are Adobe Illustrator and Corel DESIGNER.
- iii. **Photo Editing Software** allow you to enhance poor-quality photos by adjusting brightness and contrast, cropping out unwanted portions, and so on. An popular example of photo editing software is Adobe Photoshop.

## 4.0 Conclusion

---

Computer hardware cannot function on its own because software is needed to direct the operations of the computer. System software such as operating systems and utilities help the computer to perform the most basic functions while Application software interacts with system software to help the computer user perform useful tasks.

## 5.0 Summary

---

You have learnt that:

- i. The two major categories of software are system software and application software.
- ii. System software help the computer to carry out its basic operating functions.
- iii. Types of system software are operating systems, utilities and language translators.
- iv. Application software help users to do real work such as creating documents.
- v. Types of application software include word processors, spreadsheets, database management systems, presentation software and graphics software.

## 6.0 Tutor-Marked Assignment

---

1. Describe the two major categories of software.
2. Differentiate between compilers and interpreters.
3. What is an operating system? List three examples of operating systems.
4. Which type of application software would you likely use to:
  - a. Write a letter to your head of department
  - b. Store the details such as names, residential addresses, email addresses, phone numbers, etc. of your friends
  - c. Display charts about the number of students admitted in your department in the last five years
  - d. Make an oral presentation of your assignment to the whole class
  - e. Change the background of your photograph

## 7.0 References/ Further Readings

---

Parsons, J. J., Oja, D. (2011), *Practical Computer Literacy* 3rd edition, Centage Learning.  
Norton, P. (2005), *Peter Norton's Introduction to Computers* 6th edition, McGraw-Hill/Irwin  
Leon, A., Leon, M., (1999), *Fundamentals of Information Technology*, leonVikas Hall

# Unit 2

---

## Operating Systems

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Historical Evolution of Operating Systems
  - 3.2 Types of Operating Systems
  - 3.3 Functions of Operating Systems
  - 3.4 Virtual Machines
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Operating systems are the most important software that run on a computer. They manage the computer hardware, and act as a midway between the computer user and the hardware. This unit describes the history of operating systems, beginning with computers that didn't have any operating systems, and ending with operating systems used in of today's computers. Like most man-made systems, the many features of operating systems were introduced as engineers' experience accumulated over time.

Operating systems vary greatly. For example, personal computer operating systems are designed to support games, business applications and so on. On the other hand, operating systems for mainframe computers are designed to maximize the utilization of hardware.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to:

- i. Describe the history of operating systems
- ii. Explain the types of operating systems
- iii. Discuss key functions of operating systems
- iv. Describe virtual machines

## 3.0 Main Content

---

### 3.1 Historical evolution of operating systems

---

**Serial Processing:** Be aware that from the late 1940s to the mid-1950s, the earliest computers had no operating systems, thus the programmer interacted directly with the hardware. Programs written in machine were loaded via a card reader (i.e., the input device). If a program successfully completed execution, the output appeared on a printer. However, if the program halted due to an error, the error condition was indicated by some display lights. The programmer and/or computer operator had to supply all the instructions to carry out even the most basic tasks. There were two main problems with the earliest computers.

Because users had to reserve computer time, the computer remained idle if a user's program completed execution before the allocated time. On the other hand, if a user ran into problems, he/she might be forced to leave the computer once the allocated time was used up, even though the program has not completed execution.

Considerable time was needed to setup a program to run. A single program could involve loading a high-level program and compiler into memory, saving the compiled program, and so on. If an error occurred in the process, the user had to go back to the beginning of the sequence.

This mode of operation is referred to as serial processing, since users had to access the computer in series.

**Simple Batch Systems:** The first operating systems were batch systems which appeared in the mid-1950s. Batch systems used a simple operating system known as a monitor, which eliminated the need for users to directly access the processor. Instead, a user submits a

program or job to the computer operator on tape or disk. The operator queues up the jobs, and later submits an entire batch of jobs to the monitor on an input device. Each program is designed to return to the monitor upon completion, so that the monitor can load the next job. The major bottleneck of batch processing systems were input/output operation: all computation had to stop to let an I/O operation take place.

**Multi-Programmed Batch Systems:** Even though job sequencing is automatically handled in simple batch systems, the processor is often idle. The reason is that I/O devices are slow compared to the processor. A processor may spend up to 90% of its time waiting to read from/write to a file. Multiprogramming or multitasking systems allow the monitor as well as several other programs to be loaded into the computer memory. Processor idle time is minimized because while one program is performing I/O, another program can make use of the processor. This approach is used in many modern operating systems.

**Time-Sharing Systems:** Multiprogramming allows batch systems to be efficient, but it doesn't allow the user to interact with the computer during processing. User interaction is essential in certain jobs such as transaction processing. Timesharing allows the processor time to be shared among multiple users. Each user accesses the computer through a terminal. The operating system allocates a small time slice to each job, and then moves to the next job. The process is repeated until all jobs are completed. Because the time slot is very small, the operating system returns to each job after a short time, causing each user to think that only his/her job is running on the computer.

## 3.2 Types of operating systems

---

There are a wide variety of operating systems in use today. Some of these types of operating systems are:

**Mainframe Operating Systems:** Mainframes are the room-sized computers still found in major corporate data centers. Operating systems for mainframes are geared towards processing multiple I/O-intensive jobs at the same time.

**Server Operating Systems:** Server operating systems run on servers, which could be large personal computers, or even mainframes. They serve multiple users at once over a network and allow the users to share software and hardware resources. Servers can provide one or more of the following types of services: file service, print service, or Web service.

**Multiprocessor Operating Systems:** Multiprocessor operating systems are needed to help connect multiple CPUs into a single system. They differ from server operating systems, because they have special features for handling communication, connectivity, and consistency.

**Personal Computer Operating Systems:** Personal Computer operating systems provide good support to a single user. They support multiprogramming, and are widely used for word processing, spreadsheets, and Internet access.

**Handheld Computer Operating Systems:** Handheld computers include mobile phones as well as Personal Digital Assistants (PDA), which are used for performing a small number of functions, such as an electronic address book and memo pad. The operating systems that run on these handhelds have the ability to handle telephony, digital photography, emails, Internet access, and other functions.



**Embedded Operating Systems:** Embedded systems run on the computers that control devices that are not usually thought of as computers. Since they do not accept user-installed software. Examples are car engines, MP3 players, TV sets, microwave ovens, and DVD recorders. Embedded systems differ from handheld devices, because entrusted software will not run on embedded systems. For example, you cannot download new applications to your microwave oven, since all the software is already stored in the ROM.

**Sensor Node Operating Systems:** Sensors are small battery-powered computers with built-in radios. These nodes are tiny computers that communicate with each other and with a base station using wireless communication. They have limited power and must work for long periods of time unattended outdoors, frequently in environmentally harsh conditions. Sensor networks are used to detect fires in forests, measure temperature and precipitation for weather forecasting, protect the perimeters of buildings, and guard national borders, and so on. The operating system for sensor networks need to be small and simple because the nodes have little RAM and battery lifetime is a major issue.

**Real-Time Operating Systems:** Real-time systems have time as a key parameter. These systems must provide absolute guarantees that a certain action will occur by a certain time. For example, if a car is moving down an assembly line, certain actions must take place at certain instants of time. If a welding robot welds too early or too late, the car will be ruined.

### Self-Assessment Exercise(s) 1

1. The earliest computer systems which had not operating systems used a technique known as .....
2. Simple batch systems used an operating system known as a .....
3. Why is the processor often idle in simple batch systems?
4. List any three types of operating systems.

## 3.3 Functions of Operating Systems

---

The operating system acts as an interface between application software and the hardware. The key functions of an operating system include:

**User Interface:** Virtually all operating systems provide a user interface (UI) for users to interact with the computer. For example, a command line interface allows users to enter text commands, whereas a graphical user interface allows users to interact with on-screen objects and menus using a pointing device.

**Program Execution:** An operating system performs several tasks in order to execute a program: it loads the instructions and data into memory, it allocates the processor to the program, and so on.

**I/O Operations:** A running program may require I/O from a file or I/O device. In order to ensure efficiency and protection, users cannot control I/O directly. Instead, the operating system provides a means to carry out I/O.

**File System Manipulation:** Operating systems provide file systems that allow users and programs to create, list, search for, delete, read and write files and directories.

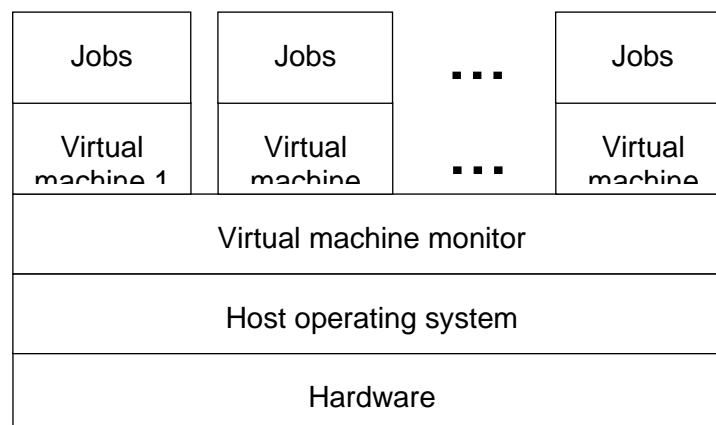
**Resource Allocation:** Operating systems handle the allocation of resources to multiple users and/or multiple jobs running at the same time. Such resources include CPU, main memory and file storage.

**Protection and Security:** Protection is concerned with controlling the access of genuine users/jobs to the computer resources. On the other hand, security involves safeguarding the computer system from outsiders. An example of protection is when the operating system prevents several jobs running at the same time from interfering with one another or with the operating system. An example of security is when the operating system requires users to validate themselves by means of passwords, in order to gain access to system resources.

### 3.4 Virtual Machines

---

A virtual machine (VM) abstracts the hardware of a single computer (such as CPU, memory and disk drives) into a number of execution environments, making it seem like each execution environment is running its own private computer. Virtualization allows a single computer to run multiple operating systems, or multiple sessions of an operating system. A host operating system can support several virtual machines. A virtual machine monitor (VMM) runs on top of the host operating system, or is incorporated into it. Each virtual machine runs a separate virtual operating system. In order to execute a job, the VMM hands over processor control to a virtual machine operating system.



**Virtual machine concept**

### 4.0 Conclusion

---

However, Computer operating systems have evolved from serial and batch operating systems to those that support multiprogramming and timesharing. Operating systems perform many important functions such as provision of user interface, program execution, and management of I/O and file operations.

### 5.0 Summary

---

You have learnt that:

- i. Computer systems initially had no operating systems, but modern operating systems support multiprogramming and timesharing.

- ii. There are different types of operating systems, such as mainframe, server systems, personal computer and embedded operating systems.
- iii. The functions of operating systems include provision of user interface, program execution, resource allocation, and ensuring protection and security.
- iv. Virtualization allows several operating systems to simultaneously run on a single computer.

## 6.0 Tutor-Marked Assignment

---

1. List the two limitations of early serial processing systems.
2. How were the above mentioned limitations tackled in simple batch systems?
3. Describe the different types of operating systems.
4. Discuss the key functions of operating systems.
5. What are virtual machines?

## 7.0 References/ Further Readings

---

- Stallings, W. (2012), Operating systems: Internals and design principles, 7<sup>th</sup>ed, Prentice Hall, New Jersey
- Tanenbaum, A. (2009), Modern Operating Systems, 3<sup>rd</sup>ed, Prentice Hall, New Jersey
- Silberschatz, A. Galvin, P. B., Gagne, G., (2010), Operating Systems Concepts, 8<sup>th</sup>ed, John Wiley & Sons, New Jersey

# Module 3

---

## Machine Level Representation of Data

- Unit 1: Units of Data and Fundamental Operations on Bits
- Unit 2: Number Bases and Types
- Unit 3: Representation of Non-Numeric Data, Records and Registers

# Unit 1

---

## Units of Data and Fundamental Operations on Bits

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Units of Information: Bits, Byte and Word
  - 3.2 Measure of Bits and Bytes
  - 3.3 Fundamental Operations of bits
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Let us recall that the main use of a computer is to process data into information. Data refers to information in an unorganized and raw form. For the data to be processed, it has to be fed to the computer and then it is processed in batches. In this class, we shall be looking at the various units of data, starting from the smallest basic unit to the largest. We shall also introduce the fundamental operations that can be performed on the smallest unit of data.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to:

- i. Define and explain the concept of bits, bytes and words
- ii. Identify and differentiate the different units of information
- iii. Perform fundamental operations on bits

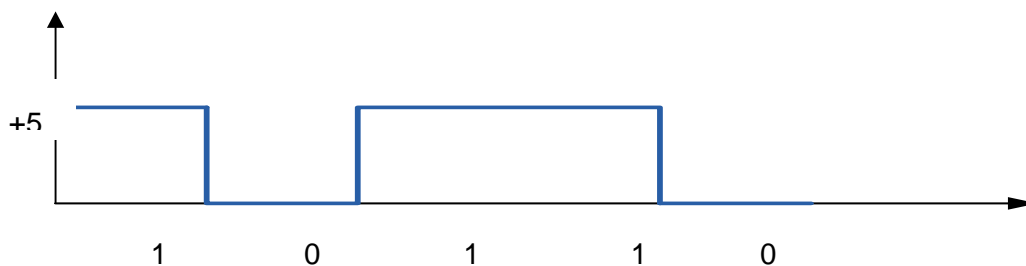
## 3.0 Learning Contents

---

### 3.1 Units of Information

---

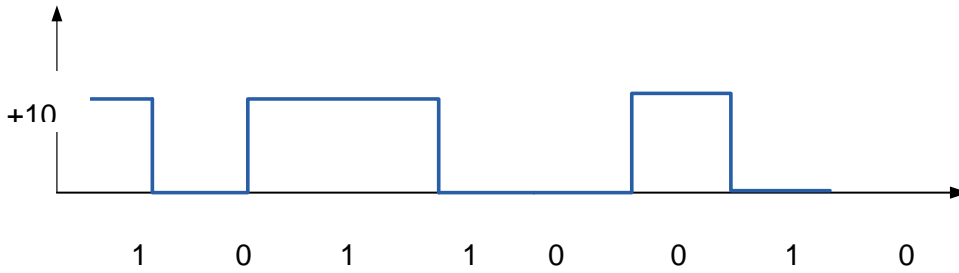
Bits are the shortened form of the term binary digit and it represents information in two states only. The two states are zeros and ones (0 and 1). The 0 represents OFF or LOW states while the 1 represents ON or HIGH states. The two states can be likened to light in a room with two options only, the lights can be ON or OFF with no state in between. Two important factors of a bit are its duration and the difference in voltage levels between the 0 and the 1.



**Example 1:** Figure showing a string of bits 10110

In example 1 is displayed a string of 5 bits. A +5v was used to represent the ON state (1) and 0v used for the OFF state (0). This is not fixed, and can be +5v for 1 and -5v for 0, +10v for 1 and +5v for 0. The duration of each bit in the example will depend on the duration of the pulse. If the pulse lasted for 10 seconds, each bit will have a duration of  $10/5 = 2$  seconds.

A group of eight bits put together is referred to as a byte. The amount of information contained increases as the number of bits increases. This is because each of the bits can be either LOW or HIGH and the different sequences of ONs and OFFs can be used to convey information. 8 bits put together can be used to represent the various characters in most languages, and this was then adopted to be 1 byte. Bits is usually represented with a lower case letter 'b', while the upper case later 'B' is used to represent a byte.



Example 2: Figure showing one byte: 1011010

A computer has the capability to process large amount of data at a time. Feeding just 8 bits (1 byte) to a computer to process at a time is a waste of processing power. Just like tying a small bucket to a horse to pull. The number of bits a particular computer can process at a particular time is referred to as word. For example, in a 32-bit computer, the word length is 32 bits and such a computer can address 32 memory locations for processing at a particular point of time. Other examples of word lengths are 64bits, 16bits etc.

Self-Assessment Exercise(s) 1

1. How many states can a bit take?.....
2. How many bits make a byte? .....
3. Represent the following graphically: 1001100111 using a +12v for ON and -12v for OFF, let the pulse last for 10seconds.

3.2 Measures of Units of Information

Computers can process and store large amount of data and information. As such terms are used to refer to this large aggregate amount of bits or bytes. Table 1.1 shows the various units of information you need to comprehend.

Name	Representation	No of Bits	No of Bytes
1kilobits	1kb	$2^{10} = 1,024$	$1024/8 = 128$
1megabit	1mb	$2^{20} = 1,048,576$	$1048576/8 = 131,072$
1gigabit	1gb	$2^{30} = 1,073,741,824$	$1073741824/8 = 134,217,728$
1terabit	1tb	$2^{40} = 1,099,511,627,776$	$1099511627776/8 = 137,438,953,472$
1Kilobyte	1KB	$8 \times 2^{10} = 8,192$	$8192/8 = 1,024$
1Megabyte	1MB	$8 \times 2^{20} = 8,388,608$	$8388608/8 = 1,048,576$
1Gigabyte	1GB	$8 \times 2^{30} = 8,589,934,592$	$8589934592/8 = 1,073,741,824$

1Terabyte	1TB	$8 \times 2^{40} =$ 8,796,093,022,208	$8796093022208/8 =$ 1,099,511,627,776
-----------	-----	--	--

**Table 1:** The different Units of Information

It is pertinent that you can comfortably transform one unit to the other.

**Example 3:** Find the number of bytes in 7mb.

To do this you will need to first find the number of bits in 7mb, which is  $7 \times 2^{20} = 7340032$  bits. You then divide by 8 to find the number of bytes as 1 byte = 8 bits. Therefore, number of bits in 7mb =  $7340032/8 = 917504$  bytes.

**Example 4:** A computer network has a download speed of 1MBps. What is the speed in mbps?

Remember that B is for Bytes while b is used to represent bits. The given speed is therefore 1 Megabytes per second, and you are required to find the speed in megabits per seconds. Since 1 byte is equivalent to 8 bits, 1MB = 8mb. Therefore, the computer download speed of 1MBps is equivalent to 8mbps.

### Self-Assessment Exercise(s) 2

- i. A computer hard disk has a capacity of 1TB. How many Mb of data can it contain?
- ii. Flash disk with a capacity of 4Gb is used to copy video files of 64Mb each. How many of the files can be stored on the disk.
- iii. An Internet Service Provider offers a download speed of 1gbps. How long will it take to download a 10Mb file? (Hint: Time = data size (in Gb) /speed)

### 3.3 Basic Operations on Bits

Be informed that during processing, computers manipulate the bits to give a desired result. This manipulation can be a complex process comprising of basic simple operations on the bits. In this section, you will learn how to carry out basic operations of addition, subtraction, division, multiplication of bits and logical operations on bits.

#### Addition of Bits

The addition of bits is similar to the addition of decimal numbers, just that a 2 is treated as a 10, and 1 added to the next bit on the left. The steps you will follow to add to bits together are outlined:

- Step 1: Align the bits to be added to the right
- Step 2: Start by adding the bits to the far right column
- Step 3: Add the bits in each column together, using the rule ( $1+0 = 1$ ,  $0+0 = 0$ ,  $1+1=0$  add 1 to the next column)



**Example 5:** Add this two bits 10011010 and 11010111

Carry bit: 1 1 11

$$\begin{array}{r}
 10010010 \\
 + \quad \underline{11010111} \\
 \hline
 \underline{\underline{101101001}}
 \end{array}$$

**Example 6:** Find the sum of these bits 10111001, 110101, 10110

$$\begin{array}{r}
 10111001 \quad -1^{\text{st}} \text{ Bits} \\
 + \quad \underline{110101} \quad -2^{\text{nd}} \text{ Bits} \\
 11101110 \quad -\text{Sum of } 1^{\text{st}} \text{ and } 2^{\text{nd}} \text{ Bits} \\
 + \quad \underline{10110} \quad -3^{\text{rd}} \text{ Bits} \\
 \hline
 \underline{\underline{100000100}} \quad -\text{Sum of } 1^{\text{st}}, 2^{\text{nd}} \text{ and } 3^{\text{rd}} \text{ Bits}
 \end{array}$$

### Subtraction of Bits

The subtraction of bits is achieved by adding the 2's complements the number to be subtracted to the other bit. The procedure for obtaining the 2's complement is: Turn all the 0 to 1, and the 1 to 0, Add 1 to the last bit on the right. After getting the complement, add to the other binary, but ignore any overflow bit. Overflow bit is the extra bit to the left after the addition.

**Example 7:** Find 1100 – 1001

Step 1: Find the 2's complement of 1001. Flipping all bits gives 0110, adding 1 to the last bit gives 0111.

Step 2: Adding 1100 to the complement 0111.

$$\begin{array}{r}
 1100 \\
 + \quad \underline{0111} \\
 \hline
 \underline{\underline{40011}}
 \end{array}$$

The last bit to the left is the overflow bit, which is ignored after the addition.

Therefore 1100 – 1001 = 0011.

**Example 8:** Find 11001 – 1110

Step 1: Find the 2's complement of 01110. Note that it was padded with a 0 to make it equal in length to 11001. Flipping all bits gives 10001, adding 1 to the last bit gives 10010.

Step 2: Adding 11001 to the complement 0010.

$$\begin{array}{r}
 11001 \\
 + \quad \underline{10010} \\
 \hline
 \underline{\underline{401011}}
 \end{array}$$

The last bit to the left is the overflow bit, which is ignored after the addition.

Therefore  $11001 - 1110 = 1011$ .

It can be observed that the bit after the overflow bit in the two examples are 0. This indicates that the result is a positive number. If it is 1, then the result is a negative number. The following example demonstrates this result of getting a negative result.

**Example 9:** Find  $10010 - 11001$ .

Step 1: Find the 2's complement of 11001. Flipping all bits gives 00110, adding 1 to the last bit gives 00111.

Step 2: Adding 10010 to the complement 00111.

$$\begin{array}{r} 10010 \\ \underline{00111} \\ \underline{11001} \end{array}$$

### Division of Bits

The division of bits is done by a series of subtractions of the divisor from the dividend, noting the remainder which is added to the other bits of the dividend. To achieve this, the following steps should be followed:

Step 1: Align the non-zero most significant bits of the divisor and the dividend.

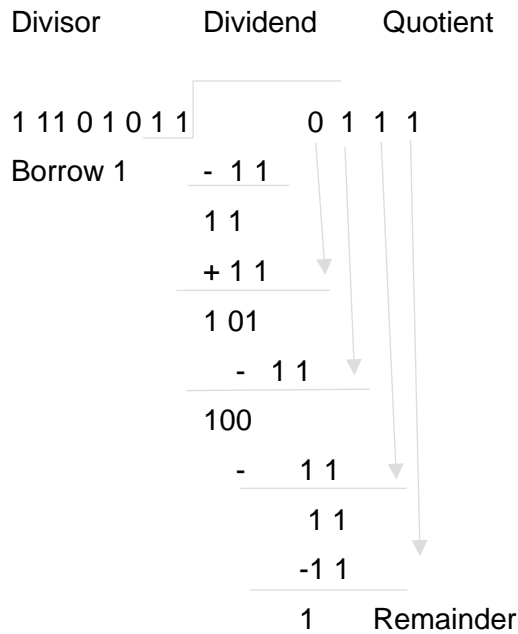
Step 2: Subtract the divisor from the most significant bits of the dividend

Step 3: if a borrow was needed, then the quotient/result is 1, and is 0 if no borrow was needed.

Step 4: If a borrow was needed in step 4, add the divisor to the result of the subtraction to restore the dividend and drop the next significant bit. If no borrow was needed in the subtraction, just drop the next most significant bit to the result of the subtraction.

Step 5: Repeat steps 2 – 4 until all the bits of the dividend have been dropped down and the divisor subtracted and the quotient updated. The quotient and remainder after the last bit is dropped is returned as the result of the division.

**Example:** find 101011 / 11



Therefore 101011 / 11 = 0111 with 1 remainder

### **Multiplication of Bits**

The multiplication of binary number can be achieved by a series of shifting and addition. This can be achieved by following these steps

- Step 1: The numbers to multiplied are aligned to the left
- Step 2: The first bit to the left of the second number is used to multiply through all the bits of the first number. Here  $1 \times 0 = 0$ ,  $0 \times 1 = 0$ ,  $1 \times 1 = 1$ .
- Step 3: You then multiply the second bit of the second number through all the bits of the first, shift the result one bit to the right, and place below the result of step 1.
- Step 4: Keep multiplying through until all the bits of the second number have multiplied the bits of the first number.
- Step 5: Sum together the results of the multiplication of each bits

**Example:** Multiply 101101 x 110101

$$\begin{array}{r} 101101 \\ \times 110101 \\ \hline 101101 \\ 000000 \\ 101101 \\ 000000 \\ + 101101 \\ \hline 1110110001 \end{array}$$

### Self-Assessment Exercise(s) 3

- Solve the following  
**a.** 0101     **b.** 01111     **c.** 111010     **d.** 1101010  
 $+ 0011 - 0101 \div 101 \times 110$
- Find the result of the addition, subtraction, multiplication and division of 1101110 and 1101

## 4.0 Conclusion

---

Computers are used to process data into information. These data are represented in Bits, Bytes and word. Bits are the basics unit of information and can exist in two states; 0 and 1. Eight bits of data put together makes a byte, while the amount of bits are computer can process at a time is known as a word. Basic operations of addition, subtraction, multiplication and division can be performed on bits.

## 5.0 Summary

---

In this unit, you have learnt:

- Data can be represented using bits, bytes and word.
- How to convert from one unit of representation to another.
- How to perform fundamental operations on bits.

## 6.0 Tutor-Marked Assignment

---

- Explain the difference between bit, byte and word.
- A computer hard disk has a capacity of 2TB. How many mb of data can it contain
- Use hexadecimal notation to represent the following bit patterns:  
**a.** 0110101011110010     **b.** 111010000101010100010111     **c.** 01001000
- Solve the following

$$\begin{array}{l} \mathbf{a.} \quad 101101 \mathbf{b.} \quad 110110 \quad \mathbf{c.} \quad 101101 \quad \mathbf{d.} \quad 10101011 \\ + 110001 \quad - 100101 \quad \div 1101 \times 1101 \end{array}$$

## 7.0 References/ Further Readings

---

Brookshear J. G. (2005), *Computer Science: An Overview*, 9th Edition, Pearson Addison Wesley.

Norton, P. (2005), *Peter Norton's Introduction to Computers*, 6th edition, McGraw-Hill/Irwin

Pfaffenberger, B. (2002), *Computers in your Future* 4th edition, Prentice Hall

Rajaraman V and Radhakrishanan T, (2009), *An Introduction to Digital Computer Design*, 5th Edition, PHI Learning Private Limited.

# Unit 2

---

## Number Bases and Types

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Base Two (Binary Numbers)
  - 3.2 Base Ten (Decimal Numbers)
  - 3.3 Base Sixteen (Hexadecimal Numbers)
  - 3.4 Integers and Floating-Point Numbers
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

In the previous unit, the bit was introduced to be the shortened form of the binary digit that can exist in two states, 1 and 0. This can be referred to as a base two number. The base of a number refers to the number of numerals or states that the number can contain. The common bases that are used to represent data to the computer system are the base 2 (binary), base 10 (decimal) and base 16 (hexadecimal). This unit will be focusing on the binary, decimal and hexadecimal number bases and the types of numbers.

## 2.0 Learning Outcomes

---

At the end of this unit, the student will be able to

- i. Express numerical data in base 2, 10 and 16
- ii. Convert numerical data from one base to the other
- iii. Identify and differentiate the various types of numbers

## 3.0 Learning Contents

---

### 3.1 Binary Numbers (Base two)

---

As previously explained, a base two number can only exist in two states, 0 and 1. Each of the numeral is referred to as a bit. Series of bits put together to represent data that have more than two states. Table 2.0 shows a string of four bits (known as a nibble, half of a byte) and the decimal equivalent of the various states.

State	Binary	Decimal Equivalent
1	0000	0
2	0001	1
3	0010	2
4	0011	3
5	0100	4
6	0101	5
7	0110	6
8	0111	7
9	1000	8
10	1001	9
11	1010	10
12	1011	11
13	1100	12
14	1101	13
15	1110	14
16	1111	15

A string of 4 bits can be used to represent 16 states of information to the computer system. This is from the fact that  $2^3 = 16$ . The base is raised to power 3 because counting the number of bits from 0 (for the least significant bit) to 3 for the most significant bit. As such each bit in base two have a value that depends on its position in the binary stream.

Power of Base	3	2	1	0	Decimal Equivalent
Binary Number	1	1	1	1	$1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16$
	1	0	1	0	$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 10$
	0	1	1	0	$0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 6$

As table 2.1 shows, conversion of binary numbers to decimal numbers involves numbering the bits from 0 for the least significant to most significant. The value of the bit is then used to multiply the base raised to power of the base. The higher the number of bits in the system, the greater the amount of states and then data that can be fed into the computer, processed and stored.

Let us treat the example below: Convert 11001010 to a decimal. How many states of information can a byte contain?

In the solution, a conversion table is filled and worked out

Power of Base	7	6	5	4	3	2	1	0	Decimal Equivalent
Binary Number	1	1	0	0	1	0	1	0	$1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 202$

For the number of states, all the bits are set to 1, that is 11111111 and when converted to decimal gives 255, but 1 is added to the 255 to compensate for the 0 state to give 256 states.

Do you understand?

### Self-Assessment Exercise 1

1. Convert  $10101110_2$  to decimal number
2. What is the decimal equivalent of  $10110001_2$ ?


## 3.1 Decimal Numbers (Base 10)

You will agree with me that decimal numbers are the most common base that are used to represent data input to computers and the output of processing are usually in decimals. This base has ten states/numerals ranging from 0 to 9. The position of the numeral also determines its value. The position 0 is referred to as unit, position 1 as tens, 2 as hundreds, 3 as thousands. For example, the number 529 has 9 units, 2 tens and 5 hundreds all added together. Since this base is quite common, much explanations will not be necessary, but the conversion of base 10 to base 2 will be concentrated upon.

The conversion of a base 10 to base 2 involves series of division of the base 10 number by 2, recording the remainders from bottom to top yields the base 2 number. As an example, let's work out the conversion of  $105_{10}$  to base 2.



	Divide	Remainder
2	105	
2	27	1
2	18	1
2	9	0
2	4	1
2	2	0
2	1	0
	0	1



Reading the remainders upwards as shown by the arrow give  $1001011_2$  as the solution.

### Self-Assessment Exercise(s) 2

1. Convert  $307_{10}$  to a base 2 number
2. What is the binary equivalent of  $296_{10}$ ?

### 3.2 Hexadecimal Numbers (base 16)

Hexadecimal numbers have 16 different states/numerals. It uses a combination of the first ten numerals of the decimal numbers and the first six alphabets. This shown in table 2.3

State	Hexadecimal	Decimal Equivalent	Binary Equivalent
1	0	0	0000
2	1	1	0001
3	2	2	0010
4	3	3	0011
5	4	4	0100
6	5	5	0101
7	6	6	0110
8	7	7	0111
9	8	8	1000
10	9	9	1001
11	A	10	1010
12	B	11	1011
13	C	12	1100
14	D	13	1101
15	E	14	1110
16	F	15	1111

A prefix of 0x before a number shows it is a hexadecimal number e.g. 0x3A7. The conversion of a binary number to hexadecimal involves dividing the binary number into a section of four bits and each nibble (4 bits' stream) converted to its hexadecimal equivalent using table 2.4.

**For example,** a conversion of 11010011100100101011 to hex will involve the following

Step 1: Segment the bit stream to 4 bits	1101	0011	1001	0010	1011
Step 2: Convert each bit to Hex	D	3	9	2	B
Step 3: Join the hex together with 0x prefix	0xD392B				

Are we together?

Conversion to decimal will involve a numbering all the numerals, starting from 0 for the least significant bit. Let us take a look at example 2.4 shows the conversion of 0xD392B to decimal.

Power of Base	4	3	2	1	0	Decimal Equivalent
Binary Number	D	3	9	2	B	$D(13) \times 16^4 + 3 \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + B(11) \times 16^0 = 866603$

The conversion of decimal to hexadecimal, just like the conversion from decimal to binary, involves series of division (this time by 16) and remainders written from the last to the first gives the hexadecimal number.

For example conversion of 734593 to hex is:

	Divide	Remainder
16	734593	
16	45912	1
16	2869	8
16	179	5
16	11	3
	0	11 (B)

From the table,  $734593_{10} = 0xB3581$ .

### Self-Assessment Exercise(s) 3

- Use hexadecimal notation to represent the following bit patterns:
  - 1100101111011010
  - 101010100101010100010111
  - 01101001
- What bit patterns are represented by the following hexadecimal patterns?
  - 7AD8
  - 71BD
  - A47D
  - 010A

### 3.3 Integers and Floating Point Numbers

---

There are a lot of different types of numbers and each is different from another, yet they may share some common characteristics. To help keep track of and understand the similarities and differences between numbers, mathematicians have developed a grouping system that categorizes and describes numbers based on their characteristics. Some of the most common groups in the system are: Natural numbers, Whole numbers, Integers, Rationals, Real numbers.

#### Integers

An integer is a whole number (is not a fractional number) that can be positive, negative, or zero. Examples of integers are: -6, 2, 8, -95, and 2015. Examples of numbers that are not integers are: -2.37,  $\frac{5}{7}$ , 3.142, .07, and 45.1. The set of integers, denoted  $Z$ , is formally defined as follows:  $Z = \{\dots, -3, -2, -1, 0, 1, 2, 3 \dots\}$ . Integers are represented in computers using two notations: two's complement notation and the excess notation, both of which are based on binary numbers.

In the two's notation, a fixed number of bits are used to represent the integers. The number of bits used determines the range of integers that the notation can represent. Today's computers use 32 bits to represent integers, but we shall be using a 4-bit system to explain how it works. The most significant bit (that is the rightmost bit is used to represent the sign of the number, a 0 is used to represent positive, while a 1 depicts a negative. The procedure for obtaining the 2's complement of a positive number is: First to get its binary equivalent, secondly flip all the bits (convert all the 0 to 1, and the 1 to 0) and finally add 1 to the last bit on the right.

Bit Pattern	Value Represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

A four-bit system can only represent the integers from +7 to -8, this is why today's computers use more bits to represent integers. For example, find how a computer would represent a -14 using its two's complement.

- Step 1: Obtain the binary equivalent of 14, this is 1101
- Step 2: Add the sign bit, 01101 (it is zero for the positive 14)
- Step 3: Flip all the bits, this results in 10010
- Step 4: Add 1 to the last bit, this results in 10011.

Therefore, the two's complement representation for -14 is 10011. To represent -14, a minimum of five bits is required.

Integers can be represented also using another method known as the excess notation. To get this notation system, the bit pattern length to be used is identified and the different binary bit patterns listed from 0000... Upwards. We then identify the point where the most significant bit changes sign, and assign 0 to that point. All bit patterns upwards of that point is positive, while those below represent negative integers. Table 2.7 shows a four-bit length used to represent integers using the excess notation, the shaded portion represents the part with 1 as the most significant bit and therefore representing the positive integers.

	<b>Excess Notation</b>	<b>Integer Represented</b>
Positive integers	1111	7
	1110	6
	1101	5
	1100	4
→	1011	3
	1010	2
Change of sign	1001	1
	1000	0
Negative integers	0111	-1
	0110	-2
	0101	-3
	0100	-4
	0011	-5
	0010	-6
	0001	-7
	0000	-8

### Self-Assessment Exercise(s) 4

1. Find the two's complement of the following.  
 a) -62      b) 18      c) -104
2. State how many bits are required to represent each of the two's complement in 1.
3. Design an excess notation system using 5 bits and show the integers it can represent.

### Floating Point Numbers

Floating point numbers refers to numbers that are not whole, but contain a part that is less than one (a fractional part). There is usually a decimal point between the part of the number greater than one and the part that is less. These numbers are represented to the computer in a form called floating point notation. The highest order bit is designated as the sign bit, a zero in this bit signifies a positive number, while a 1 is used to represent a negative number. The remaining bit is divided into two parts, the exponent (used to hold the part greater than one) and the mantissa (which holds the part less than one). For example, in an eight-bit system, the first bit is used as a sign bit, the next 3 are the exponent and the last four used to hold the mantissa.

For example, if a floating point notation contains 01011101, what is the binary number represented? What is the decimal equivalent of the notation?

Break up the given bit stream to the three segments

Sign Bit	Exponent	Mantissa
0	101	1101

The given stream represents a positive bit stream of 101.1101.

To obtain the decimal equivalent, a conversion to decimal using the table below is necessary

	Sign bit	Exponent			Mantissa			
<b>Binary Number</b>	0	1	0	1	1	1	0	1
<b>Power of Base</b>		2	1	0	-1	-2	-3	-4
<b>Decimal Equivalent</b>	+	$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = 5.8125$						

Therefore, the bit stream 101.1101 represents 5.8125.

### Self-Assessment Exercise (s) 5

1. What is the binary number and decimal equivalent of the following floating-point notations?  
 a) 11010001  
 b) 01110101

The representation in the binary notation can introduce error into the system due to the limited number of bits used to hold both the exponent and the mantissa. This error is known as truncation error or round-off error. This error is reduced by using more bits to represent the mantissa field and in today's computers use at least 32 bits for storing values in the floating point notation instead of the 8 bits used in the examples.

## 4.0 Conclusion

---

Numeric data needs to be represented in the computer in a form that it can be processed to give information. This can usually be done using the base 2, 10, 16 notation depending on the circuitry of the processor of the computer system. This unit introduced the three common bases numbers are represented in the computer and also how integers and floating point numbers are handled.

## 5.0 Summary

---

In this unit, you have learnt:

- i. Binary number representation of numeric data and conversion from base two to base ten.
- ii. Decimal number representation of numeric data and conversion from base 10 to base 2
- iii. Hexadecimal number representation of numeric data and conversion from base 16 to base ten and base two.
- iv. Representation of Integers and Floating point numbers.

## 6.0 Tutor-Marked Assignment

---

1. Convert each of the following binary representations to its equivalent base ten form:  
**a. 0101 b. 1001 c. 1011 d. 0110 e. 10000 f. 10010**
1. Convert each of the following base ten representations to its equivalent binary form:  
**a. 6 b. 13 c. 11 d. 18 e. 27 f. 4**
2. Use hexadecimal notation to represent the following bit patterns:  
**a. 0110101011110010 b. 111010000101010100010111 c. 01001000**
3. What bit patterns are represented by the following hexadecimal patterns?  
**a. 5FD97 b. 610A c. ABCD d. 0100**
6. Find the two's complement of the following.  
a) -5    b) 28    c) -172

## 7.0 References/ Further Readings

---

- Brookshear J. G. (2005), *Computer Science: An Overview*, 9th Edition, Pearson Addison Wesley.
- Norton, P. (2005), *Peter Norton's Introduction to Computers*, 6th edition, McGraw-Hill/Irwin
- Pfaffenberger, B. (2002), *Computers in your Future* 4th edition, Prentice Hall
- Rajaraman V and Radhakrishanan T, (2009), *An Introduction to Digital Computer Design*, 5th Edition, PHI Learning Private Limited.

# Unit 3

---

## Representation of Non-Numeric Data, Records and Registers

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Non-Numeric Data and Representation
  - 3.2 Computer Records and Registers
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

In the previous unit, number bases were introduced to be the number of numerals or states that the number can contain. The common bases that are used to represent data to the computer system are the base 2 (binary), base 10 (decimal) and base 16 (hexadecimal). This unit will be focusing on the representation of non-numeric data such as alphabets and special characters, and also records and registers will be introduced.

## 2.0 Learning Outcomes

At the end of this unit, the student will be able to

- i. Differentiate numeric data from non-numeric data
- ii. Explain the various ways non-numeric data is represented in computers
- iii. Identify and itemize the functions of records and registers.

## 1.0 Learning Contents

### 3.1 Representation of Non-numeric data

Non-numeric data refers data that are not numbers and cannot be arranged serially based on their values. These cannot be added nor subtracted like numeric data and examples include alphabets, special characters, images and sound. The number bases explained earlier can be used to represent numeric data to computers, but the representation of non-numeric data makes use of a coded format. The coded format used for each type is different and can be varied. This unit will focus on the representation of alphabets and special characters.

The American National Standards Institute (ANSI) came up with the American Code for Information Interchange (ASCII) which have been adopted widely and used to code the representation of letters. In this code seven bit are used to represent the upper and lower cases alphabets, punctuation marks and other control functions on the keyboard such as the carriage return, tabs and backspace. Show in Table 3.1 is the ASCII representation for alphabets and special characters. From the sequence shown, the codes for the English letters are in the same sequence as their verbal order. As such the codes for alphabets can be determined if the codes for the lower and upper case of 'A' are known.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL



Table 3.1: Table of ASCII Codes

**Example:** What will be the ASCII code (in binary) to represent CODE!

From the table, you can obtain the decimal code for each of the characters of the word CODE!. Then convert each of the decimal code to binary. Note that the case of the letter matters as uppercase codes differs from lower case codes.

<b>Letter</b>	C	O	D	e	L	!
<b>Decimal</b>	67	79	68	101	108	33
<b>Binary</b>	1000011	1001111	1000100	1100101	1101100	0100001

Therefore, the ASCII code for CODE!. Is 100001110011111000100110010111011000100001.

The ASCII code is limited in the number of bits used in the coding. As such cannot be used to represent the alphabets of languages that have a large number of alphabets such as Chinese and Japanese. A more recent coding system known as the Unicode have been developed by leading computer manufacturers which uses 16 bits to represent each symbol, resulting in 65,536 unique different pattern that can handle languages with large number of alphabets. The International Organization for Standardization have also developed a code format that uses 32 bits for its encoding. This will be able to represent billions of different symbols.

### Self-Assessment Exercise(s) 1

1. What is the ASCII code for FUT, Minna
2. What does the following ASCII code represent: 1001000 1100101 1101100 1101100 1101111 0101110 (Hint: First convert each segment of seven binary code to decimal, and look up the character it represents on the table).

## 3.2 Computer Records and Registers

Have you noticed that generally, a record is a combination of various data objects such as some integers, floating-point numbers, and character strings that are arranged for processing by a program. A file contains multiple records and can also be referred to as a data set. The programming language and/or the application used determines the organization of data in the record. Typically, records can be of fixed-length or be of variable length with the length information contained within the record.

A register is a memory that is built into the central processing unit to speed up its operations by providing access to commonly used values. They are usually of small value and can be accessed very quickly. They are made of semiconductor devices whose contents can be read and written to at extremely high speeds but the contents are held only temporarily, usually while in use or only as long as the power supply is on.

Registers are the fastest way for the system to operate data and are at the top of the memory hierarchy. Registers are normally measured by the number of bits they can contain. Examples include an 8-bit register (can store 8 bits of data), a 32-bit register (can store 32 bit of data). There are basically two kinds of registers, a set of 32 registers that is accessible to a

programmer (known as the General Purpose Registers) and another set of 6 registers not accessible to programmers that is used in interpreting and executing instructions. Data and instructions must be put into the system. So we need registers for this. The basic computer registers with their names, size (for an 8085 microprocessor) and functions are listed below, the size indicate can vary depending on the processor in question.

Register Symbol	Register Name	Number of Bits	Description
AC	Accumulator	16	One of the Processor Register that holds operands and the result of arithmetic operations
DR	Data Register	16	Hold memory data
TR	Temporary Register	16	Holds temporary Data
IR	Instruction Register	16	Holds the current instruction code being execute
AR	Address Register	12	Holds memory address
PC	Program Counter	12	Holds address of next instruction, after executing the current instruction
INPR	Input Register	8	Holds Input data
OUTR	Output Register	8	Holds Output data

## 4.0 Conclusion

---

Non numeric data refers to raw information that are not numbers and cannot be added and subtracted. These include alphabets and special characters. Representation of non-numeric data in computers is done using a coded form. Various codes are used, common examples include the ASCII and the Unicode. Computer records are combinations of various data objects such as some integers, floating-point numbers, and character strings that are arranged for processing by a program, while a register is a memory that is built into the central processing unit to speed up its operations by providing access to commonly used values.

## 5.0 Summary

---

In this unit, you have learnt:

- i. Binary number representation of numeric data and conversion from base two to base ten.
- ii. Definition of computer records
- iii. Definition of computer register and some examples of computer registers.

## 6.0 Tutor-Marked Assignment

---

1. How many bits are used to code special characters using ASCII code? How many characters can be coded using ASCII?

2. Encode these sentences in ASCII character code:
  - a. "Come!" Dr. Musa said
  - b. Does  $4+3=7$ ?
2. What are computer records?
3. What are computer registers?
4. List five examples of computer registers and state their functions.

## 7.0 References/ Further Readings

---

Brookshear J. G. (2005), *Computer Science: An Overview*, 9th Edition, Pearson Addison Wesley.

Norton, P. (2005), *Peter Norton's Introduction to Computers*, 6th edition, McGraw-Hill/Irwin

Pfaffenberger, B. (2002), *Computers in your Future* 4th edition, Prentice Hall

Rajaraman V and Radhakrishanan T, (2009), *An Introduction to Digital Computer Design*, 5th Edition, PHI Learning Private Limited.

# Module 4

---

## Logic

- Unit 1: Digital Logic
- Unit 2: Memory
- Unit 3: Von Neumann Model of Computation

# Unit 1

---

## Digital Logic

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Concept of Digital Logic
  - 3.2 Combinational and Sequential Logic Circuit
  - 3.3 Introduction to Boolean Algebra and Logic Gates
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

You would agree with me that a large proportion of electronic equipment, including computers uses digital electronics where the quantities (usually voltage) are described by two states (on and off). These two states can also be represented by true and false, 0 and 1 but in most physical systems they are represented by the voltages 5V and 0V or something close to that. The operation of almost all modern digital computers is therefore, based on two-valued or binary system.

This unit introduces you to the fundamental concept behind all modern computer system and the basic motivation and wisdom behind the design and application of various digital/computer components and systems. The concept of Combinational logic and Sequential logic are presented. The unit also presented the introduction to the primitive logic devices called Logic Gate which form the building block of all digital logic circuit design.

## 2.0 Learning Outcomes

---

At the end of this unit, you should be able to achieve the following Objectives:

- i. Describe the fundamental concept behind the design of various computer/digital hardware component and systems.
- ii. Differentiate clearly between Combinational logic circuit and Sequential logic circuit.
- iii. Conceptualize the basic operations and different types of Logic Gates.

## 1.0 Learning Contents

---

### 3.1 Concept of Digital Logic

---

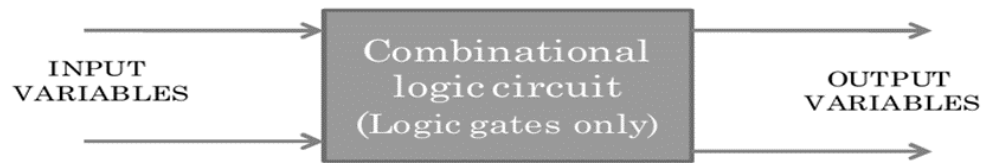
Digital logic is simply a system of rules that allow us to make extremely difficult decision based on relatively two states questions, and therefore, formed the basic motivation and rationality underlying the design and application of various computer hardware components and systems. Boolean algebra (due to George Boole) is the mathematics of digital logic and is useful in dealing with binary system of numbers. Boolean algebra is used in the analysis and synthesis of logical expressions and for manipulation of object that can take only two values typically true or false or any pair of values. The principles of Boolean algebra are applied to switches, providing mathematical tools for analysis and synthesis of any switching system. The application of Boolean algebra in design of modern computing system and devices is very significant and obvious today. This is because the relationship between Boolean logic and physical component of any computer system is very strong. Each hardware component of a computer/digital system is built of several logic circuits. Digital logic circuit is classified into two categories namely: Combinational logic circuit and Sequential logic circuits. Details of each category is given in the next sub-section.

### 3.2 Combinational and Sequential Logic Circuit.

---

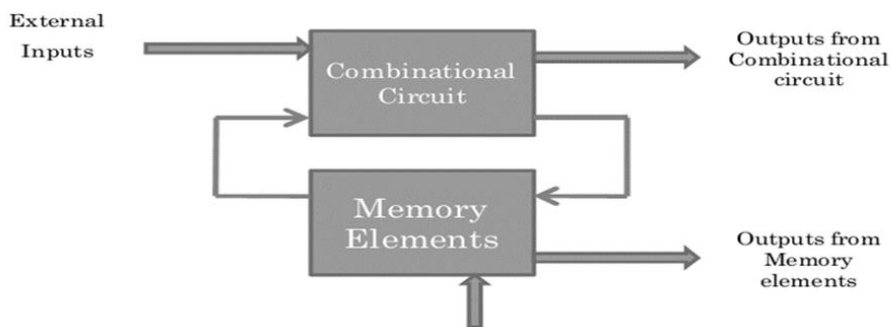
A logic circuit is known to be an interconnection of several primitive logic devices called *Logic Gates* to perform a desired function. When logic gates are connected together to give a specified output for certain specified combination of input variables, with no storage involved, the resulting network is known as Combinational logic circuit. In other words, a combinational

logic circuit contains only Logic gates but does not contain storage elements. The output of a combinational logic circuit at all the time is a pure function of the present input level only. Figure 1.0 shows you the block diagram of combinational logic circuit.



**Figure 1.0:** Block diagram of combinational logic circuit

However, a sequential logic circuit is a combinational logic circuit with memory element(s). It consists of a combinational circuit to which memory elements are connected to form a feedback path. In contrast to the combinational logic circuit, the output of a sequential logic circuit depends not only on the present input but also on the history of the input. In other words, sequential logic has memory while combinational logic does not. Figure 1.2 shows the block diagram of a sequential logic circuit.



**Figure 1.2:** Block diagram of a sequential logic circuit

It can be seen from the diagram that the output(s) of sequential logic circuit is (are) dependent not only on external input(s) but also on the present state of the memory element(s). The next state of the memory element(s) is also dependent on external input and the present state.

Although, every digital system is likely to have combinational circuits, most systems encountered in practice also include storage elements, which require that the system be described in terms of sequential logic. Practical computer circuits normally contain a mixture of combinational and sequential logic. For example, the part of an arithmetic logic unit, or ALU, that does mathematical calculations is constructed using combinational logic. Other circuits used in computers, such as half adders, full adders, half subtractors, full subtractors, multiplexers, de multiplexers, encoders and decoders are also made by using combinational logic. To implement a sequential circuit, a storage device is required to recognize what has happened in the past. The basic unit of storage is called flip-flop or Latch and the simplest kind of sequential circuit is a memory cell that has only two states. Such two state sequential circuits are called flip-flops, bistable multivibrator, latch or toggle.

## Self-Assessment Exercise(s) 1

1. Digital logic is a system of rules that allow us to make extremely complicated decision based on relatively two states questions (True/False)
2. A logic circuit is an interconnection of several primitive logic devices called ..... to perform a desired function
3. A combinational logic circuit contains both logic gates and storage elements. (True/False)

### 3.3 Introduction to Boolean Algebra and Logic Gate

---

Boolean algebra is the natural way to represent digital information since computers/digital devices are built as a collections of switches that have only two states (either “on” or “off”). As introduced earlier in 3.0, Boolean algebra, the mathematics of digital logic is used in the analysis and synthesis of logical expressions.

Logic circuits are usually represented by Logical expressions which is a combination of logical variables ( $x, y$ ) and logical operators (AND, OR, NOT, etc.).

A Boolean function typically has one or more input values and yields a result, based on these input values in binary form  $\{0, 1\}$ . The three basic Boolean operations/operators are: OR, AND, and NOT operation.

Other two Boolean operators (NOR and NAND) are combination of two of this three basic operators as shows below:

NOR = NOT and OR

NAND = NOT and AND

Thus, the five most common Boolean operators/operations are:

**(OR, AND, NOT, NOR, and NAND) operators.**

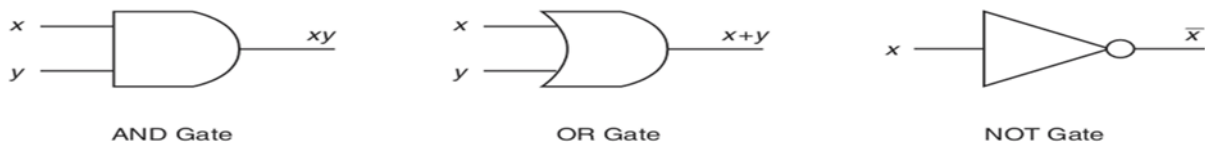
A Boolean operator can be completely described using a truth table that lists the inputs, all possible values for these inputs, and the resulting values of the operation for all possible combinations of these inputs. A truth table shows the relationship, in tabular form, between the input values and the result of a specific Boolean operator or function on the input variables.

**A logic gate** is a small electronic device that implements a simple Boolean function. They make the logical decisions based on the logic operations and became the basic building blocks for digital circuit design. Each of the five Boolean operations described above has its equivalent Logic Gate which is the actual physical electronic device that makes the logical decisions based on the corresponding logic operation. Thus, the five equivalent Logic Gates are:

- i. OR Gate which perform OR operation
- ii. AND Gate which perform AND operation
- iii. NOT Gate which perform NOT operation.
- iv. NOR Gate which perform NOR operation
- v. NAND Gate which perform NAND operation.



The graphical representation of the three basic logic gates and their equivalent true tables are respectively shown in figure 1.3 and 1.4 below:



**Figure 1.3:** Logic gates graphical representation

Inputs		Outputs
x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Outputs
x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1

Inputs	Outputs
x	$\bar{x}$
0	1
1	0

Figure 1.4: Truth Tables for AND, OR and NOT logic operations

**AND Operation:** AND operation is represented by  $xy = x \cdot y$ . Its associated Truth Table is shown in fig 4. A truth table gives the value of output variable (here xy) for all combinations of input variable values. Thus in an AND operation, the output will be 1 (True) only if all of the inputs are 1 (True).

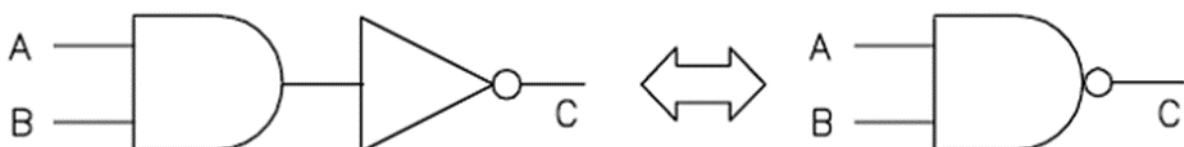
**OR Operation:** OR operation is represented by  $x \text{ OR } y = x + y$  where x and y are logical (Boolean) variables and the + sign represents the logical addition called an 'OR' operation. Its associated Truth Tables are shown in fig 4. Thus in an OR operation, the output will be 1 (True) if either of the inputs is 1 (True). If both inputs are 0 (False), only then the output will be 0 (False).

Note that though, the symbol + is used, the logical addition described above does not follow the rules of normal arithmetic addition.

**NOT Operation:** The NOT gate has only one input which is then inverted by the gate. The symbol and truth table for the operation are shown in fig 1.3 and 1.4.:

**NAND Gate:** As introduced to you earlier, we could combine AND and NOT gates together to form a NAND gate. Thus the logical expression for a NAND gate is  $C = \overline{A \cdot B}$

The symbol and truth table are shown in figure 1.5. The NAND gate symbol is given by an AND gate symbol with a circle at the output to indicate the inverting operation.



A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

Figure 1.5: logic symbol and Truth Table for NAND Gate

**NOR Gate:** Similarly, OR and NOT gates could be combined to form a NOR gate. The Gate symbol and Table are depicted in figure 1.6.

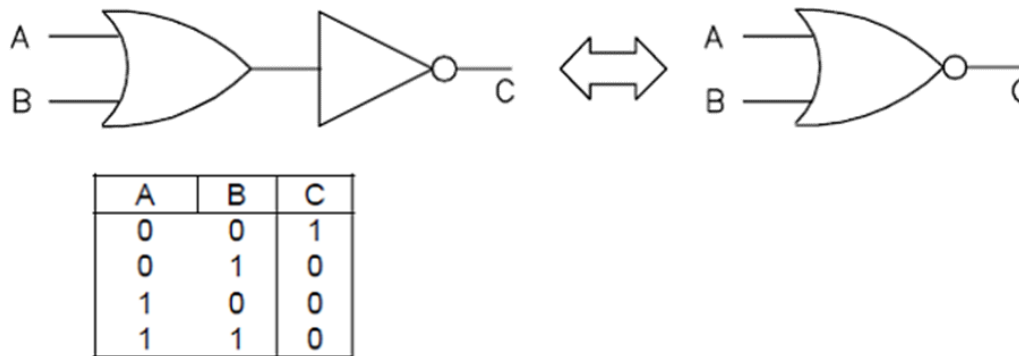


Figure 1.6: logic symbol and truth table for NOR Gate

### Self-Assessment Exercise(s) 2

1. An actual physical electronic device that makes logical decisions based on the corresponding logic operation is called .....
2. The three basic Boolean operations/operators are: .....
3. Boolean algebra is the natural way to represent digital information (True/False)

## 4.0 Conclusion

In this unit, you have learnt the fundamental concept that behind the design of various computer/digital component and system which is Digital Logic. You have also been able to understand the concept and the building blocks of combinational and sequential logic circuits. These building blocks which are small digital devices (i.e. Logic Gates and Storage elements) are the basic building elements of digital components circuit design. If we must understand the basic logic gates operations, then I believe you have learnt the fundamental concept of digital/computer hardware and system design.

## 5.0 Summary

Many scientific and electronic equipment, including computers uses digital electronics where the quantities (usually voltage) are described by two states. Consequently, the operation of almost all modern digital computers system and devices is based on two-valued or binary system. In this unit, we have gone through the concept of Digital/Boolean Logic which form

the basis of any digital or computer component and systems. You have seen that the digital logic circuit is sub-divided into Combinational logic circuit and Sequential logic circuit. To ensure you appreciate the basic concept, we introduced to you, the standard logical operators and looked at the Logic Gates that implement them. The interconnection of several Logic Gates without storing element to perform a desire function defined combinational logic circuit. Whereas, a Sequential logic circuit is a combinational logic circuit with storing element(s). You have also seen that Boolean algebra is the mathematic of digital logical and is used in the analysis of logical expressions. Logic circuits are usually represented by Logical expressions which is a combination of logical variables (x, y) and logical operators (AND, OR, NOT, etc.).

## 6.0 Tutor Marked Assignment

---

1. What do you understand by the term Digital Logic?
2. Clearly differentiate between Combinational logic circuit and Sequential logic circuit
3. Describe the term Logic Gate.

## 7.0 References.

---

- The Essentials of Computer Organization and Architecture. Linda Null and Julia Labur. 2003.
- Computer Organization, Design, and Architecture. Fourth Edition. Sajjan G. ShivaCRS Press.
- Stallings, W. Computer Organization and Architecture, 5th ed., New York: Macmillan Publishing Company, 2000.
- Computer Organization and architecture. Patterson and Hennessy (1997)
- hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach, 2nd ed., San
- Gregg, John. Ones and Zeros: Understanding Boolean algebra, Digital Circuits, and the Logic of Sets. New York: IEEE Press, 1998.
- Tanenbaum, Andrew. Structured Computer Organization, 4th ed. Upper Saddle River, NJ: PrenticeHall, 1999.

# Unit 2

---

## Memory

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Memory Concept
  - 3.2 Types of Memory
  - 3.3 Memory hierarchy
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

We have introduced to you the term “Flip-flops” in this module as the basic unit of storing binary information in sequential logic circuit. A register is also a device used either to store data temporarily or to influence data stored in it using the logic circuitry around it. A set of such registers where programs and data are stored is called Memory. This unit introduces you to the concept of Memory subsystem of a digital computer. The two basic classification of memory devices is presented. They are Read only memory (ROM) and Random access memory (RAM). The unit also presents cache memory and the concept of Memory hierarchy then followed.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to achieve the following Objectives:

- i. Define the term “Memory”
- ii. Differentiate between Primary and Secondary memory
- iii. Differentiate between RAM and ROM
- iv. Identify the significant of Memory Hierarchy in modern computer system.

## 3.0 Learning Contents

---

### 3.1 Memory Concept

---

The term Memory subsystem of a digital computer is simply a set of registers where programs and data are stored. It is logically structured as a linear array of locations, with addresses from 0 to the maximum memory size the system processor can address.

It is required, for best performance of the machine that, the stored data and programs be accessible by control and processing units as fast as possible. The main memory described as primary memory permits such a fast access. This fast-access requirement extend a significant amount of hardware to the main memory and thus makes it expensive. Data and programs that are not needed immediately by the machine are usually stored in secondary memory to reduce memory cost. They are brought into the main memory when need by processing unit. We will examine various types of memory in the next sub section.

### 3.2 Types of Memory

---

Although, you may see different classification of memory devices depending on different mechanism. The two basic types of memory devices are:

Random-access Memory (RAM) and Read-only Memory (ROM)

**Random Access Memory (RAM):** Random Access simply means the content of data and programs stored in the memory can be accessed in a random manner. In this type of memory, any addressable location can be accessed randomly. In other words, the method of

'writing into' and 'reading from' a location in a RAM is identical and consumes the same time regardless of its position in the memory.

The terms *Read/Write memory (RWM)* and *RAM* are used interchangeably in most literature to describe the same thing. RAM is the most common type of Primary/main memory in utmost digital system/devices. The specification of 500 megabytes RAM of your computer simply refers to 500MB of RWM. Although, RAM stores data and programs the computer needs when executing instructions, it is volatile, and loses this information once the power is turned off. Each memory register or location in a RAM is associated with an "address". Data are 'written into' and 'read from' RAM memory location by using address to locate the position of the data in the memory. There are generally, two types of semiconductor RAMs based on the chips used to build the bulk of RAM memory in today's computer systems: they are Static and Dynamic RAM (SRAM and DRAM). They can be differentiated as follow:

- i. Static RAM is constructed with a flip-flop while Dynamic RAM is constructed out of capacitors that leak electricity.
- ii. Consequently, the contents of SRAM (either 1 or 0) is static as long as the power is on whereas, the state of 1 or 0 is determined by the charge level of the capacitor in dynamic RAM.
- iii. Since the charge decays with time, the memory cell of DRAM requires often refresh (i.e. recharge) to hold its memory content.
- iv. Because of the refresh time required by DRAM, they are slower and less expensive than SRAM.
- v. However, DRAM uses less power, generates less heat and more DRAM cells can be fabricated on the same area of silicon than SRAM cell.
- vi. Thus, designers often use both technologies in combination: DRAM for main memory and SRAM for cache memory.

Although, the basic operation of all DRAM memories is the same, different types of DRAM exists as listed below:

- i. Multibank DRAM (MDRAM),
- ii. Fast-Page Mode (FPM) DRAM,
- iii. Extended Data Out (EDO) DRAM,
- iv. Burst EDO DRAM (BEDO DRAM),
- v. Synchronous Dynamic Random Access Memory (SDRAM),
- vi. Synchronous-Link (SL) DRAM,
- vii. Double Data Rate (DDR) SDRAM, and
- viii. Direct Rambus (DR) DRAM.

SRAM also exist in different types as listed below:

- i. Asynchronous SRAM,
- ii. Synchronous SRAM, and

- iii. Pipeline burst SRAM.

**Note:** The basic operation of all SRAM memories is also the same and not discuss in this course material.

**Read-Only Memory (ROM):** Remember in RWM, Data are 'written into' and 'read from' the memory location by using address to locate the position of the data in the memory. The characteristics of ROM are as follow:

- a. Although, ROM is also a random access memory, Data can only be 'read from' the memory location of ROM.
- b. Unlike RAM, data are usually 'written into' the ROM either by memory manufacturer or by the user using special devices that can write (burn) the data pattern into the ROM.
- c. In most computers, ROM is a smaller memory in addition to RAM that stores data and programs necessary to boot computer system.
- d. ROM is not a volatile memory and is used in any system that does not require alteration in the system programming.
- e. Many appliances, calculators, peripheral devices and most automobiles use ROM chips to store and preserve information when the power is shut off.

There are five basic different types of ROM: (1) ROM (2) PROM (3) EPROM, (4) EEPROM, and (5) flash memory.

- 1) **PROM** (Programmable Read-Only Memory) is a variation on ROM that can be programmed by the user with an appropriate equipment. While ROMs are hardwired, PROMs have fuses that can be blown to program the chip. Data and instructions in PROM cannot be altered once programmed.
- 2) **EPROM** (Erasable PROM) is a PROM with the added advantage of being reprogrammable with the aid of a special tool that emits ultraviolet light.
- 3) **EEPROM** (Electrically erasable PROM) eliminates shortcomings of EPROM by applying an electrical field that can erase only portions of the chip, one byte at a time.
- 4) Flash Memory is Basically an EEPROM with the additional advantage that data can be written or erased in blocks, eliminating the one-byte-at-a-time limitation. This makes flash memory faster than EEPROM

### Self-Assessment Exercise(s) 1

- 1. A set of registers where programs and data are stored in digital computer system is called .....
- 2. Data and programs not immediately needed by the machine are usually stored in ..... memory to reduce memory cost.
- 3. Data can only be 'read from' the memory location of RAM. (True/False)
- 4. The terms Read/Write memory (RWM) and RAM are used interchangeably (True/False)
- 5. Data are usually 'written into' the ROM either by memory manufacturer or by the user using a special devices that can write (burn) the data pattern into the ROM. (True/False)

### 3.3 Memory Hierarchy

---

Let us recall that computer processor usually and frequently reading information from computer memory and the processor speed is higher than the memory access time. The primary memory of a computer system is constantly built out of RAM devices, thereby allowing the system processing unit to access data and instructions in the memory as quickly as possible. Since primary memory devices are more expensive than secondary memory devices, a cost effective memory system results when the primary memory capacity is minimized. To reduce the cost and the speed gap, a small but faster memory called cache memory is usually introduced between the main memory and the processing unit.

Cache memory is simply a small, temporary, and very high-speed memory that the processor uses for information it is likely to need again in the very near future. It is usually faster than the primary memory.

In a cost effective practice, it is not necessary to store the whole data or program in the primary memory as long as the portion of the program or data needed by the processing unit is in the primary memory. A secondary memory built out of direct- or serial -access devices is then used to store programs and data not immediately needed by the processing unit. Modern computers usually have cache memory where data from frequently used memory locations may be temporarily stored.

Hence, memory hierarchy is one of the most significant considerations in understanding the performance capabilities of a modern processor. However, as we have introduced to you so far, memories are differently created with different characteristics and capability. Some types are far less efficient and therefore cheaper than others. To overcome this difference, a combination of memory types is used in today's computer systems to achieve the best performance of the system at the lowest cost. This methodology is called Memory Hierarchy.

Thus, in a general-purpose computer system, there is a Memory Hierarchy in which the highest speed and most expensive memory device is closest to the processing unit while the slowest and least expensive memory devices are farthest from the processing unit.

The memory system hierarchy shared in most today's computer systems comprises of the following levels:

1. CPU registers (basic unit of storage)
2. Cache memory (a fast, small RAM block)
3. Primary (main) memory (the RAM from which the processor accesses all programs and data via the cache memory) and
4. Secondary (mass) memory.

**Note:** The registers in the processing unit are temporary storage devices and they are the fastest components of the computer system memory.

The speed and cost per bit of the memory is highest at level 1 and decreases across the levels to the lowest at level 4.

The main objective of the hierarchical memory design is to facilitate a speed-cost compromise that will provide a memory system of desired capacity with the highest speed possible at the lowest cost.



## Self-Assessment Exercise(s) 2

1. To reduce the cost and the speed gap, a small but very high-speed memory called ..... is usually introduced between the main memory and the processing unit.
2. Hierarchical memory design facilitates a speed-cost compromise that will provide a memory system of desired capacity with the highest speed possible at the lowest cost. (True/False).

### 4.0 Conclusion

---

In this unit, you have learnt the concept of memory subsystem. You have been able to understand the two types of memory which are *primary* and *secondary memory*. You have also been able to understand the two basic categories of memory devices which are RAM and ROM. If we must understand the concept of *memory hierarchy*, then I believe you have learnt the concept of memory subsystem of a digital computer system.

### 5.0 Summary

---

The term “Memory” of a digital computer is simply a set of registers where programs and data are stored. In this unit, we have gone through the concept of memory subsystem. You have seen that memory devices are basically grouped into is basically into Random-access Memory (RAM) and Read-only Memory (ROM): Each of the group is further grouped into static and dynamic memory depending on the semiconductor that built it. Data can be ‘written into’ and ‘read from’ the memory location of RAM whereas, Data can only be ‘read from’ the memory location of ROM. Primary memory of a computer system is constantly built out of RAM devices, thereby allowing the system processing unit to access data and instructions in the memory as quickly as possible. Data and programs not immediately needed by the machine are usually stored in secondary memory and brought into the primary memory when needed by the processing unit. Today’s computer systems used combination of memory types to achieve the best performance of the system at the lowest cost. Thus, we have finally seen that in a general-purpose computer system there is a Memory Hierarchy in which the most expensive memory device with highest speed is closest to the processing unit. This is to facilitate a cost-speed compromise that will provide a highest speed memory system of desired capacity at the lowest cost.

### 6.0 Tutor Marked Assignment

---

1. Define the term “Memory”
2. Differentiate between the following terms:
  - a. Random-access Memory and Read-only Memory
  - b. Primary memory and Secondary memory
3. Highlight the concept of Memory Hierarchy and state it is significant in today’s computer systems

## 7.0 References

---

The Essentials of Computer Organization and Architecture. Linda Null and Julia Labur. 2003. Computer Organization, Design, and Architecture. Fourth Edition. Sajjan G. Shiva CRS Press.

Stallings, W. Computer Organization and Architecture, 5th ed., New York: Macmillan Publishing Company, 2000.

Computer Organization and architecture. Patterson and Hennessy (1997)

Hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach, 2nd ed., San

Gregg, John. Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets. New York: IEEE Press, 1998.

Tanenbaum, Andrew. Structured Computer Organization, 4th ed. Upper Saddle River, NJ: Prentice Hall, 1999.

# Unit 3

---

## Von Neumann Model of Computation

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Von Neumann Model
  - 3.2 Fetch-Decode-Execute cycle
  - 3.3 Base Machine Organization
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

You would agree with me that the primary function of a computer system is to process the input data to yield a results that can be useful in a specific application environment. In 1880, Herman Hollerith developed numerous unit record machines to process data on punched cards for census applications in the United States. In 1944, the first automated computer (Mark I), that used paper tape for program storage and punched cards for input and output data was announced. The need for faster computations than those provided by Mark I, give rise to the development of the first electronic computer (ENIAC), which employed stored-program concept' by a team led by Eckert and Mauchly. In ENIAC, a sequence of instructions called 'program' is stored in the memory for use in processing data by the machine. The programs were wired on a control board and a rewiring of the board is required for each computation sequence. The first stored-program computer (EDVAC), was developed by John von Neumann, a member of the Eckert-Mauchly team in 1945. This unit introduces you to the concept of the first stored-program computer called von Neumann Model of Computation. The von Neumann architecture and Fetch-Decode-Execute cycle of the architecture are also presented in this unit. We finally examined the basic machine or computer organization.

## 2.0 Learning Outcomes

---

At the end of studying this unit, you should be able to achieve the following Objectives:

1. Recognize the von Neumann architecture as the fundamental concept of all stored-program computers.
2. Define Fetch-Decode-Execute cycle.
3. Describe the basic computer organization

## 3.0 Learning Contents

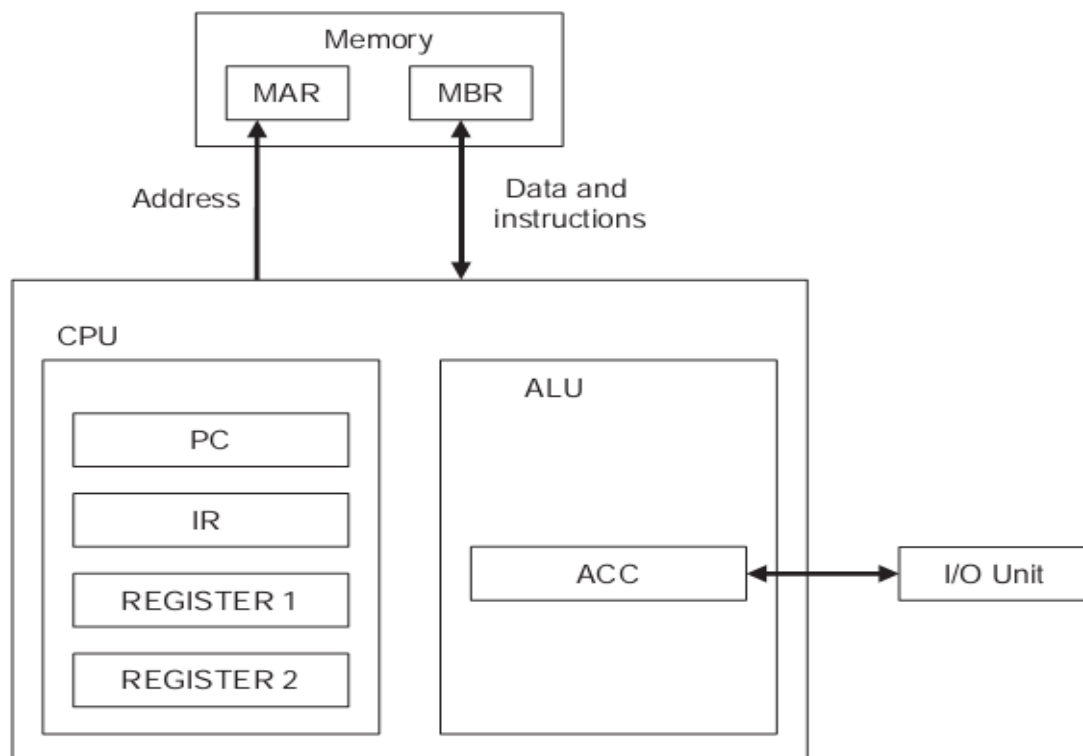
---

### 3.1 Von Neumann Model

---

The first stored-program computer (EDVAC), was developed by John von Neumann, a member of the Eckert-Mauchly team in 1945. All stored-program computers using the von Neumann architecture are known as von Neumann machine/system. Although, several different architectures have evolved, all modern computers are Von Neumann machine. As shown in Figure 3.0 below, Von Neumann model is a uniprocessor computer system that consist of the following four (4) basic units:

- i. Memory Unit; which carries Memory Buffer Register (MBR) and Memory Address Register (MAR) or Memory Data Register (MDR).
- ii. Control Unit; which contains Program Counter (PC) and Instruction Registers (IR))
- iii. Arithmetic and Logic Unit (ALU); which carries Accumulator (ACC)
- iv. Input/output Unit.



**Figure 3.0** Von Neumann architecture

**Remember:** Registers are small unit of storing data whereas, ACC is one form of register that store intermediate result of arithmetic and logic operation.

**Memory Unit:** This is a single port storing device in which the arrangement of memory cells is in form of several memory words. Each memory word is the unit of data that can be 'read' from or 'written' to the memory device through the single port of the memory.

**Arithmetic and Logic Unit:** This unit executes the arithmetic and logic operations on the data in ACC and/or MBR and the result of the operation is retained in the ACC.

**Control Unit:** The Program Counter (PC) in this unit holds the address of the instruction to be fetched whereas, the instructions are fetch into the Instruction Registrar (IR) from memory for execution. Other two included registers can be used to hold data and address values during computation.

**Input/output Unit:** This unit connects Input/output devices to the system. In this model, the I/O devises send and receive data into and from the ALU subsystem for simplicity. In practice, the data exchange may also occur directly between I/O devices and memory without exploiting any register.

It is very necessary for you to note the following characteristics of the von Neumann that make it inefficient:

- i. Data and programs are stored in a single sequential memory, which create a single path memory access referred to as the ***Von Neumann Bottleneck***.
- ii. There is no distinction in representation of data and instruction in the memory.

- iii. In data representation, there is no distinction between a set of bits representing different data e.g. floating-point data and a character string.
- iv. High level language programming environments utilize several data structure and one dimensional single sequential memory requires linearization of such data structure

Consequently, the von Neumann model requires excessive mapping by compilers to generate the executable code from the programs written in high-level languages. Despite these deficiencies, practical structure of most digital computers follows the von Neumann model. However, the development of numerous compilers over years has significantly reduced the semantic gap problem created by the model.

The von Neumann architecture described to you so far, runs programs in what is known as von Neumann execution cycle which is also called *fetch-decode-execute cycle*. Section 3.1 presents the execution cycle.

### 3.2 Fetch-Decode-Execute Cycle

The fetch-decode-execute cycle describes how the von Neumann machine operates. The following steps described one iteration of the cycle.

- a. The program counter determines the location of the instruction in the memory and the control unit *fetches* the next program instruction from the memory.
- b. The instruction fetched is *decoded* into a language that ALU can understand.
- c. All data operands necessary for execution of the instruction are fetched from the memory and booked into registers within the CPU.
- d. The ALU then *executes* the instruction and places the results in memory or register

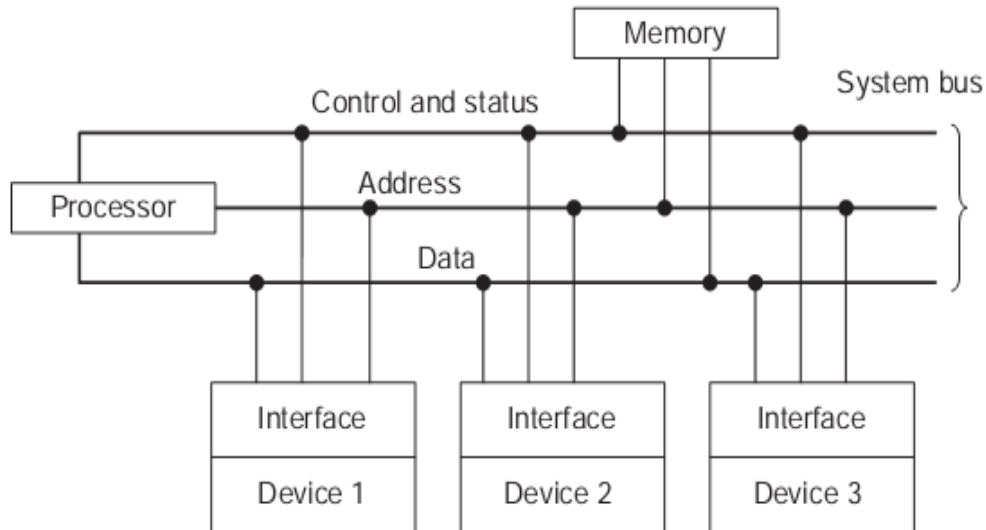
The three main operations of the cycle are *fetch*, *decode*, and *execute*. The control unit fetches the instruction from the memory, the instruction is decoded into ALU language and then finally execute by the ALU. The next sub-section presents the basic machine organization.

#### Self-Assessment Exercise(s) 1

1. The first stored-program computer, was developed by .....
2. Von Neumann model is a uniprocessor computer system that consist of four (4) basic units (True/False)
3. The control unit executes the instruction and places the results in memory or register (True/False)
4. The three main operations of the von Neumann execution cycle are .....

### 3.3 Basic Machine Organization

The main hardware components (CPU, Memory, and I/O) of most computers available in market today are interconnected according to the von Neumann model and therefore, described as von Neumann computers. However, the architecture of the model have been extended and rationalized into what is presently designated as "*system bus model*" which is shown in figure 3.1. The model shows a more generalized architecture of modern day computer system.



**Figure 3.1:** General Computer System Organization

The model architecture is made up of three major sub-systems (Processor, Memory and Input/output sub systems) which are all interconnected by the system bus. The system bus consists of three different buses which are: control bus, address bus and data bus whereas, the processor subsystem (CPU) comprises of the ALU, the Control unit and various processor registers.

Remember: The architecture described to you is a single-bus system architecture which may differ from practical system that may be configured around multiple buses to allow simultaneous operations on the buses. For example: in two bus structure, the processor may be connected to memory sub-system by memory bus whereas, the I/O bus interface the I/O devices to the processor. Although, the system complexity increases with multiple bus, there is possibility of higher throughput compared to single-bus architecture due to simultaneous operations on the buses. Consequently, a cost-speed trade-off is necessary to choose the system structure.

## 4.0 Conclusion

---

In this unit, you have learnt the concept of von Neumann Model of computation which is the fundamental concept of all stored-program computers/machines. You have been able to understand the architecture and operation cycle of von Neumann model. The architecture highlighted the hardware composition of the model whereas, the operation cycle which is called fetch-decode-execute cycle described how the von Neumann machine operates.

If we must understand the model and the basic machine organization, then I believe you have learnt the fundamental concept of all stored-program computers and machines.

## 5.0 Summary

---

The primary function of a computer system is to process the input data to yield a results that can be useful in a specific application environment. In 1880, Herman Hollerith developed numerous unit record machines to process data on punched cards for census applications in

the United States. The first stored-program computer was developed by John von Neumann, a member of the Eckert-Mauchly team in 1945 and all stored-program computers using the von Neumann architecture are known as von Neumann machine/system. In this unit we have gone through the von Neumann Model of computation which is the fundamental concept of all stored-program computers and machines. You have seen that the von Neumann architecture consists of four basic units which are: Memory Unit, Arithmetic and Logic Unit, Control Unit and Input/output Unit. To ensure you comprehend the basic concept, we described to you the “fetch-decode-execution cycle” which describes how the von Neumann machine operates. The three main operations of the cycle are fetch, decode, and execute. The control unit fetches the instruction from the memory, the instruction is decoded into ALU language and then finally execute by the ALU. Finally, you have also seen that the major hardware components of most computers available in market today are interconnected according to the von Neumann model and therefore, described as von Neumann computers. However, the architecture of the model has been extended and rationalized into what is presently designated as “system bus model”.

## 6.0 Tutor Marked Assignment

---

- i. Explain the four basic units of von Neumann architecture.
- ii. Describe the term “fetch-decode-execution cycle”
- iii. What did you understand by the term “system bus model”?

## 7.0 References

---

- The Essentials of Computer Organization and Architecture. Linda Null and Julia Labur. 2003.
- Computer Organization, Design, and Architecture. Fourth Edition. Sajjan G. ShivaCRS Press.
- Stallings, W. Computer Organization and Architecture, 5th ed., New York: Macmillan Publishing Company, 2000.
- Computer Organization and architecture. Patterson and Hennessy (1997)
- Hennessy, J. L., & Patterson, D. A. Computer Architecture: A Quantitative Approach, 2nd ed., San
- Tanenbaum, Andrew. Structured Computer Organization, 4th ed. Upper Saddle River, NJ: Prentice Hall, 1999.



# Module 5

---

## Programming the Computer

Unit 1: Programming Languages

Unit 2: Algorithms

Unit 3: Flowcharts

# Unit 1

---

## Overview of Programming languages

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Programming Languages
  - 3.2 Language Translator
  - 3.3 Examples of High Level Language
  - 3.4 Programming Paradigms
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Historically, in the early period, computers are given instructions through the control buttons. However, the buttons are fixed and not flexible for executing different instructions. Therefore, this unit will expose you to the flexible way of instructing computer machines through what is called programming language. The unit will define the programming language and its purpose; the programming paradigms; and the programming translators that are used to convert one programming language to another.

## 2.0 Learning Outcomes

---

At the end of this unit, you will be able to:

1. Define a program
2. Differentiate between the three main programming languages
3. Identify the different program translator
4. Differentiate between procedural programming and object-oriented programming

## 3.0 Learning Contents

---

### 3.1 Programming Languages

---

When we want to communicate with one another, a language is normally used either English language or any other language. Similarly, a programming language is a way of communication between a computer programmer and a computer itself. A programmer is the person that writes a set of instructions using a programming language to instruct the computer to perform a task. A program is a set of instructions that are logically related.

As we all know, every language comprises a collection of words and symbols, which the users of the language use for communication. The collection of words and symbols is known as vocabulary of a language.

However, programming languages use *syntax* instead of vocabulary. This *syntax* is a set of rules used by programmers to write programs that the computer understands.

Programming languages can be broadly categorized into three languages:

1. Machine language
2. Assembly language
3. High level language

If you recall a program written in any programming language involves a set of instructions that are logically related. These instructions have two parts, as shown in figure 1.1 below.

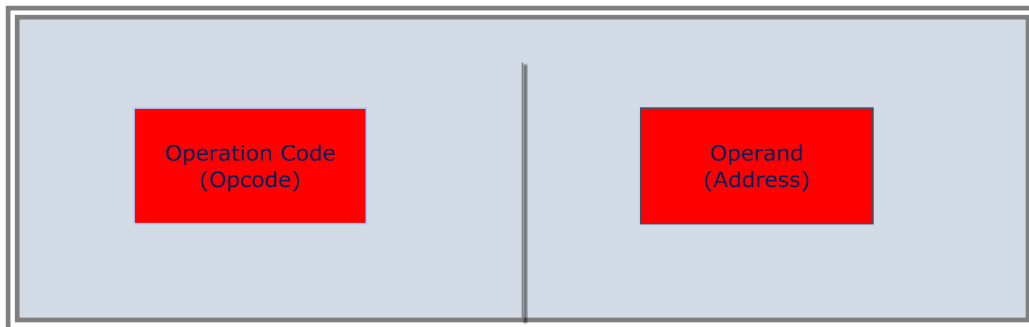


Figure 1.1 Format of a programming language instruction

The two parts of a programming language instruction are:

1. **Operation Code (Opcode):** This is the part that instructs a computer about the operation to be performed.
2. **Operand:** This is the part that instructs the computer about the location of the data on which the operation stated by the opcode is to be performed.

For example, look at this instruction Add A and B, “Add” is the Opcode while A & B are the operands.

Every computer has its own set of opcode. The following operations can be performed by the opcode in the instruction set. These operations include:

- i. **Logical Operations:** These operations are used for comparing data and the results can be either true or false.
- ii. **Arithmetic Operations:** these operations are used for mathematical calculations.
- iii. **Branch Operations:** These operations involve transfer of data to a memory location given in an operand field
- iv. **Data Movement Operations:** These operations involve the transfer of data between the memory and the processor or input/output devices and the computer.

## Machine Language

All computers use binary number system that comprise of 0 and 1 for performing internal operations.

The first language available for modern computers was the machine language and it is considered as the lowest level language. Machine language consists entirely of ONs and OFFs denoted as 1s and 0s respectively and it varies from one type of machine to another. Since computers are designed to only understand 0s and 1s, it makes execution of machine language fast because it does not require any translator.

Although, programs can be written in a number of programming languages but computer only understands machine language. Therefore, the programs written in other languages need to be translated to machine language for execution.

The machine language is difficult to read and understand because it is written in binary digits which is difficult to learn. For example, a program to store a number might be

1001000101110001

1100101000111011

Machine language is the language of computer and it is also called a first-generation language.

### **Advantages of Machine Language**

1. Program written in machine language executes fast
2. it does not require translator
3. it requires a relatively small space

### **Disadvantages:**

1. it uses binary or Hex code which are difficult for human to read
2. the programs of machine language are difficult to debug and maintain
3. The programmer needs to remember the memory locations of the data
4. Programs written in machine language are difficult to modify because of their complexity
5. The machine language code differs from one type of computer to another.

### **Assembly Language**

*Assembly language* was introduced in 1952 in order to overcome the limitations of machine language. The assembly language uses alphanumeric notations instead of binary digits to represent instructions and memory addresses. For example, it uses ADD for addition and SUB for subtraction.

The alphanumeric code uses a set of letters and numbers to represent instruction. These alphanumeric code and symbols make the program shorter and easier to write than machine language. The possibility of errors is also reduced and the program can easily be modified.

An assembly language is also called a second generation language.

Below is a sample of a program written in assembly language to add two numbers:

1. LD A<sub>x</sub>, 8
2. LD B<sub>x</sub>, 9
3. ADD A<sub>x</sub>, B<sub>x</sub>
4. LD (100), A<sub>x</sub>
5. JMP B<sub>x</sub>
6. HLT

From here, A<sub>x</sub> and B<sub>x</sub> are registers. These registers are memory locations inside a microprocessor. Every instruction and data needs to be loaded from main memory to registers before the CPU can process it.

Each line of the above program is explained as follows:

1. Loads register A<sub>x</sub> with the value, 8.
2. Loads B<sub>x</sub> with the value, 9.
3. Adds the value of register B<sub>x</sub> to the value of register A<sub>x</sub>.
4. Stores the value of register A<sub>x</sub> in the main memory location, 100.

5. Uses JMP to jump to register B<sub>x</sub> transfer the control to register B<sub>x</sub>.
6. Stops the program execution

As we have mentioned earlier, a computer cannot understand any language except machine language. Because of this, a translation program is needed to interpret assembly language into machine language code. This translation program is called an assembler. A code written in any programming language other than machine language is known as source code. This source code requires translation for execution.

### Function of an Assembler

The assembler takes each program statement in the source code of assembly language to generate a corresponding bit stream (a series of 1s and 0s of a given length). The set of instructions in a bit stream is the output of the assembler and it is called the *object code*. It is the object code that will now be executed whenever required. If errors are found in the source code, the assembler generates a list of errors instead of object code.

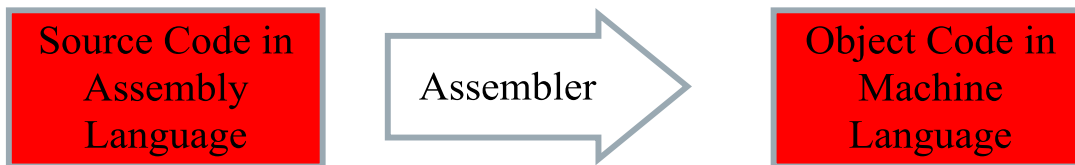


Figure 1.2 Functioning of an Assembler

### Advantages of Assembly Language:

1. Assembly language programs are easier to write and understand than machine language
2. The programmer can easily remember the opcode and alphanumeric addresses of data and instruction
3. It saves time in developing and modifying programs in assembly language

### Limitations of Assembly Language

1. The execution of programs written in assembly language is slow compare to those written in machine language
2. Programs written in assembly language depend on processors. Therefore, it cannot be transported from one machine to another

### High Level Language

To have proper understanding of machine and assembly languages are known as low-level languages because of their dependent on machine hardware. In order to create programs that are hardware independent, a new type of programming language called the *high level language* was developed by programmers. A high level language is close to human and it is user-friendly because it uses natural languages like English language to represent an instruction. It uses words such as PRINT to print and GOTO to jump to a particular line. Examples of high level language are Basic, COBOL, and Pascal.

High level language is also known as third-generation language. Each instruction in a high-level language is translated into multiple machine-level instructions.

## Advantages of High-level languages

1. High-level language is easier to learn than machine and assembly language
2. High-level language programs written in one type of computer can easily be used on another computer because it is machine independent
3. It makes reading, writing and maintaining of a program written on high-level language easy because high-level language vocabularies are similar to English language

### Self-Assessment Exercise(s) 1

1. List three categories of programming languages
2. .... is the language computer understands
3. Which of the following is correct about assembly language when compared to machine language?
  - a. Assembly language is a machine-dependent language
  - b. Assembly language programs are difficult to write
  - c. Assembly language programs have greater number of errors
  - d. Programs written using assembly language are faster

## 3.2 Language Translators

---

If we must understand, program translator plays an important role in computer programming to translate high level language in to machine language. Any program written in high level language is called a source program which is the translated into machine code (i.e. 1 & 0 codes) for machine to be able to execute the program. This conversion is done by a translator. There are two basic types of translators namely: compiler and interpreter. The following subsections will explain to you how compilers and interpreters work and also differ from each other.

### Compiler

For you to execute a program written in high-level language, a computer needs translation software called a compiler. Compiler is a program developed to convert the entire source program in to machine code at once. A compiler is always language-specific, meaning that each high-level language has its own compiler. Compiler translates the source code of that language into the object code. For example, compiler for Pascal can only compile the programs written in Pascal. It cannot compile programs written in any other language, such as COBOL. Once the compiler creates the object program, it will never use the source program again unless modification is required in the source program. It is this object file that you normally copy from one computer to another computer for you to install and use.

## Interpreter

There are some high-level languages you need to know that use different type of translator program called an interpreter. Interpreter is a program that converts the source program into a machine code. Instead of creating an executable file, interpreter translates the source code line– by- line that is what makes it run slowly compared to other translators.

Debugging of errors is easier using an interpreter because error messages in the interpreter are usually specific and points out line on which the error occurs.

PERL, BASIC, and Visual Basic languages typically use interpreter.

## Differences between a Compiler and an Interpreter

The differences between a compiler and an interpreter are explained in the table below

S/N	Differences between a Compiler and an Interpreter	
	Compiler	Interpreter
1	A compiler translates the entire program first, and generates the object code, which can be later executed	An Interpreter translates and executes one line of source code at a time
2	The object code generated after compilation is saved for future use. The programmer does not need to compile the program again for object code generation. This makes the program execution fast	The translated lines of the code cannot be saved for future use. therefore, every time code needs to be translated before execution. This makes the program execution slow
3	A compiler detects and shows syntax errors that relate to the entire source code	An interpreter detects and shows errors that relate to a line of source code

**Table 1.1** Differences between a Compiler and an Interpreter

## 3.3 Some Examples of High-level Languages

There are a number of high-level languages that have been developed since the first high-level language (FLOWMATIC) was developed in 1952. Some of the common high-level languages are Beginners All Purpose Symbolic Instruction Code (BASIC), Formula Translation (FORTRAN), Common Business Oriented Language (COBOL), Pascal, and C.

Let us take a close look at them one after the other;

### BASIC

Basic was developed in 1964 by Professor John Kenny and Thomas Kurtz at Darmouth College in the USA.

Basic is an interpreter-based language that is easy and simple to use. BASIC was one of the earliest high-level languages to be implemented on personal computers.



## **FORTRAN**

FORTRAN is one of the oldest high-level languages. It was developed in 1957 by John Backus and his team at International Business Machines (IBM) Corporation. FORTRAN was designed to solve scientific and engineering problems. After the version was released, other versions evolved, such as FORTRAN II, FORTRAN IV, and FORTRAN 77. FORTRAN IV was the first language that was standardized by American National Standards Institute (ANSI).

ANSI is a non-profit making organization, which administers and coordinates standardization and conformity assessment system in the US.

## **COBOL**

COBOL was initially developed by the mathematician, Grace Hopper, in 1959. It was later modified in 1959-60 by a committee of the conference on Data Systems Languages (DODASYL). This committee defined the language specifications for COBOL.

COBOL was standardized by ANSI in 1968. Revised versions of this standard were published in 1974 and 1985, which are known as COBOL 74 and COBOL 85, respectively. Currently, the latest version of COBOL is known as COBOL 2002. COBOL was designed to solve business data processing problems.

## **Pascal**

Pascal was standardized by ANSI in 1971 by Professor Nicklaus Wirth of the Federal Institute of Technology in Zurich, Switzerland. He named it after the famous mathematician, Blasé Pascal.

Pascal was designed to help beginners to learn good problem solving and programming practices. Today, Pascal is recognized as a language that allows programmers to write structured and modular programs. It is suitable for scientific as well as business applications.

Pascal was standardized by ANSI in 1983. Since then, different versions of Pascal are also available.

## **C Language**

C was developed in 1972 by Dennis Ritchie and Brian Kernighan at AT& T's Bell Laboratories, USA. The C was designed in such a way that it has features of a high-level language along with efficiency of low-level language.

C was standardized by ANSI in 1989. C is a compiler-based language. Therefore, C programs can be easily transferred to other computers equipped with a C compiler. It is widely used for writing word-processing programs, compilers and operating systems. For example, UNIX operating system is written in C.

## **3.4 Programming Paradigms**

---

Talking about the paradigms it is the processes involved in programming are usually accompanied by different school of thoughts called programming paradigms. They are the alternative approaches to programming processes and they also tell us how best we can program a computer. The most common important programming paradigms are procedural paradigm and the object-oriented paradigm

## Procedural Programming

In Procedural programming, a list of instructions is used to tell the computer what to do step-by-step. Procedural programming relies on procedures, also called subroutines or routines. A procedure contains a series of computational steps to be carried out. Procedural programming is also referred to as imperative programming. Examples of procedural languages include FORTRAN, COBOL and C. If you want a computer to do something, you should provide step-by-step instructions on how to do it.

## Object-Oriented Programming (OOP)

Object-oriented programming is another approach to problem-solving where all computations are carried out using objects. An **object** is a component of a program that knows how to perform certain actions and how to interact with other elements of the program. Objects are the basic units of object-oriented programming. A simple example of an object would be a person. Logically, you would expect a person to have a name. This would be considered a property of the person. You would also expect a person to be able to do something, such as walking. This would be considered a method of the person. Examples of object-oriented programming are C++, Visual Basic etc.

### Self-Assessment Exercise(s) 2

1. A translator that converts the whole program at once is called .....
2. A compiler is:
  - a. A high-level language,
  - b. Language-specific,
  - c. A machine language,
  - d. A source code
3. An interpreter is a:
  - a. Machine Language,
  - b. Program
  - c. Debugger
  - d. Translator

## 4.0 Conclusion

---

This unit presents to you an overview of programming languages. You learnt that programming languages are broadly classified in to the machine, assembly and high-level language. The machine language is the lowest programming language which is the only language computer understands and any programs written in other languages need to be translated in to machine language to enable execution of instructions by computer. It was also learnt in this unit that, there are different school of thoughts on how programs can be implemented known as programming paradigms. The most important programming paradigms are procedural and object-oriented programming.

## 5.0 Summary

---

In this unit, you learned that:

- A computer program can be written in any of the following categories of programming languages. These include: Machine language, assembly language and high-level language.
- Machine language uses the binary digits, 0 and 1 to write an instruction.
- Assembly language uses letters, numbers, and symbols instead of binary digits to represent instructions and memory addresses.
- Assemblers are programs that translate assembly language programs into equivalent machine language programs.
- High-level language programs are machine-independent
- A compiler is language-specific, and each high-level language has its own compiler that translates the program into the object code.
- An interpreter takes one instruction of a high-level language at a time, converts it to a machine language instruction and then executes it.
- Some of the common high-level languages are BASIC, FORTRAN, COBOL, Pascal, and C.

## 6.0 Tutor-Marked Assignments

---

- i. Define machine language
- ii. Give the advantages and disadvantages of a machine language
- iii. Define Assembly Language
- iv. Give the advantages and disadvantages of an assembly language
- v. Give advantages and disadvantages of a high-level language
- vi. Differentiate between a compiler and an interpreter
- vii. Differentiate between procedural paradigm and object-oriented paradigm

## 7.0 References/Further Readings

---

J. Glenn Brookshear “*Computer Science: An Overview*” *Ninth Edition* (2007)  
NIIT Student Guide: Fundamentals of Programming, Mastermind Series

# Unit 2

---

## Algorithms

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Definition of Algorithm
  - 3.2 Characteristics of a Good Algorithm
  - 3.3 Representing an Algorithm
  - 3.4 Pseudo-Code
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Remember, you have learnt in the above unit that a programming language is designed to allow human and computer to communicate in order to solve a problem. However, before writing a problem's solution in programming language, the solution to the problem needs to be designed using a tool or technique generally known as *Algorithm*. Therefore, this unit will introduce you to the concept of problem-solving methods in relation to Algorithm. The unit will also tell you the properties of a good algorithm, various forms of representing, particularly pseudo-code method of representing algorithm.

## 2.0 Learning Outcomes

---

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

- i. Define an algorithm
- ii. List the characteristics of an algorithm
- iii. Represent an algorithm using pseudo-code
- iv. Solve simple problems by using pseudo-code

## 3.0 Learning Contents

---

### 3.1. Definition of Algorithm

---

An *algorithm* is a sequence of steps required to accomplish a task. The term algorithm has no root meaning. According to the *American Heritage Dictionary*, it is derived from the name of Muhammad ibn Musa al-Khwarizmi (780A.D-850A.D), a Muslim mathematician whose works introduced Arabic numerals and Algebraic concepts to Western mathematics. The word "algebra" stems from the title of his book *Kitab al jabrwa'l-muq'abala*. In your daily and professional activities, you also perform a sequence of steps to accomplish any given task. For example, to watch a football match in a cinema, you need to first pay for your ticket and then watch the football in the cinema. You cannot directly enter the cinema hall without buying the ticket.

The *concept* meaning of algorithm has come to be accepted as: set of step-by-step instructions for solving a well-defined problem which takes allowable inputs to produce desired output. These definitions require that an algorithm must consist of:

1. Set of steps that can be clearly followed (unambiguous), typically by a computer when implemented in computer language.
2. Executable means that the steps are doable
3. Terminating process means that the process must lead to an end
4. Algorithm takes acceptable data as input.
5. It produces desired output based on the input provided. If wrong input is provided, algorithm will not give correct output.

The concept of algorithm is depicted in figure 2.1 starting from Problem definition; algorithm transformed to computer program that computer executes acceptable inputs from user to give desired output.

The concept in figure 2.1 assumes the algorithm is implemented in a programming Language. One important concept in algorithm is that the problem solver must write explicitly all the instructions required to solve the problem without assumptions.

In fact, if a problem solver cannot express the concept clearly, the algorithm will not be useful to solve the problem.

Consider a problem to find the largest of three numbers. Almost anyone can immediately tell which is the largest, but many cannot explain the steps they followed to arrive at it. Most people will exclaim, "I just know it! However, an algorithm approach desires the steps to be clearly and unambiguously stated. This is what separate algorithm from ordinary human intuition.

The algorithm is stated below.

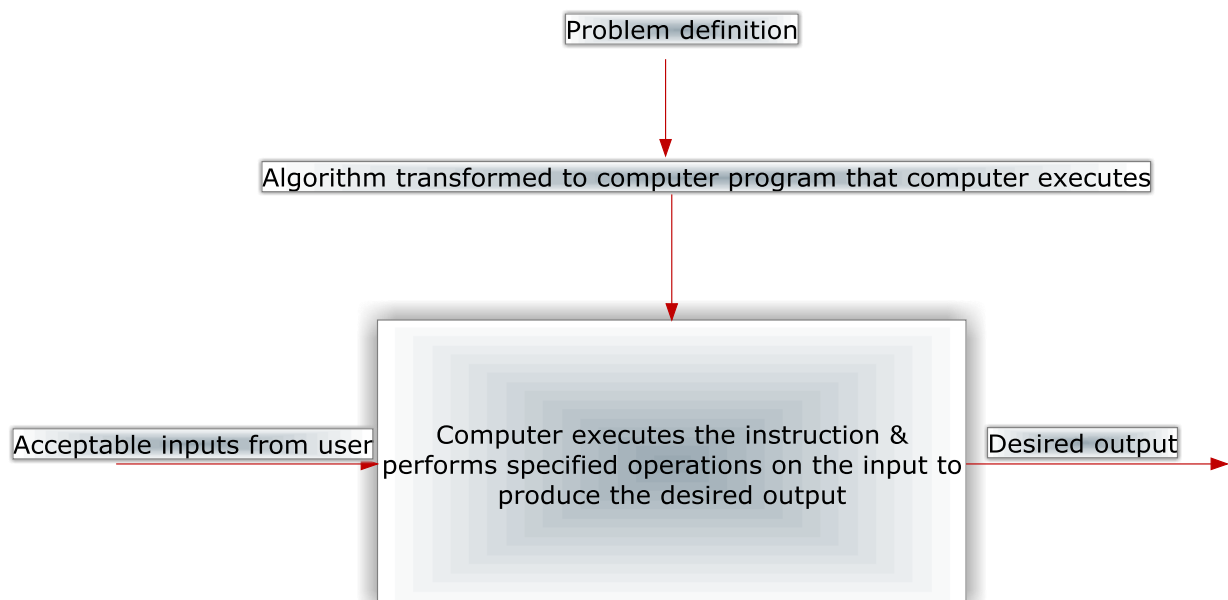
Step 1 Enter the three numbers say x,y,z.

Step 2 If  $x > y$  then go to step 3 else go to step 4

Step 3 is  $x > z$  then x is largest go to step 5 else z is largest go to step 5

Step 4 is  $y > z$  then y is the largest.

Step 5 stop



**Figure 2.1:** Concept of Algorithm

### 3.3 Representing an Algorithm

---

Algorithms can be represented in different ways, using flowcharts or pseudo codes.

**Flowcharts:** Flowcharts are graphical representation of algorithms

**Pseudo-codes** represent algorithms by simple English language.

### 3.2 Characteristics of an Algorithm

---

- i. An algorithm ends after a fixed number of steps
- ii. Each step in an algorithm clearly specifies the action to be performed
- iii. The steps in an algorithm specify basic operations. These operations could include mathematical calculations, input/output functions, and logical comparisons
- iv. The algorithm should accept input data, in a defined format, before it can be processed by the given instructions
- v. An algorithm generates one or more outputs after the input is processed. The resultant information termed as output can be displayed or stored for future reference.

#### Self-Assessment Exercise(s) 1

1. .... Is the step where solution to a problem is designed
2. List two ways of representing algorithms
3. Which of the following is not a type of algorithm?
  - a. Program,
  - b. Flowchart
  - c. Pseudo-code

### 3.3 Pseudo-Code

---

Pseudo-code represents an algorithm in the English language. It is used as an alternative approach to a flowchart. Pseudo-code uses simple statements that are written in a sequential manner.

Instead of using graphics or table to represent an algorithm, you can also use simple structure which consists of simple statements called pseudo-code. After you develop pseudo-code, you can now write it in a programming language. You can easily detect errors when pseudo-code.

The following are a number of key words used in pseudo-code.

- Begin..... end: these are used to start and finish pseudo-code. Begin is the first line and end is the last line of a pseudo-code.
- Accept: this keyword is used to receive an input from a user.
- Display: This keyword is used to present a result or an input.
- If ..... Else: These keywords are used in decision making. `

#### Examples of representing algorithm using pseudo-code

1. The following pseudo-code accepts two numbers, calculates the sum of the numbers and display the result:

```
Begin
    Accept the first number
    Accept the second number
    Calculate the sum of the first number and second number
    Display
End
```

2. The following pseudo-code accepts two numbers and displays the greater number:

Begin:

```
Accept the first number
Accept the second number
If the first number is greater than the second number
    display the "the first number is greater"
else
    display "the second number is greater"
```

end

**Other examples of algorithms we interact with in our day to day activities include but not limited to the following;**

- i. A cooking recipe booklet
- ii. The rules of how to play a game
- iii. DVD player instructions
- iv. Description of a martial arts technique
- v. Directional maps for driving from point A to B
- vi. A car repair manual

Self-Assessment Exercise(s) 1

1. List three characteristics of an Algorithm
2. Define pseudo-code
3. Write pseudo-code to find the mean of the sum of n set of numbers



## 4.0 Conclusion

---

This unit has introduced you to the concept of algorithm. The application of algorithm in solving problems was explained. The unit also shows you how to identify a good algorithm and emphasized representation of algorithm in pseudo-code. Simple problems were analyzed using pseudo-code.

## 5.0 Summary

---

1. Algorithm is a set of step-by-step instructions for solving a well-defined problem which takes input to produce desired output.
2. Algorithm can be represented using flowcharts, decision tables, or pseudo-code.
3. There are a number of properties that make any step-by-step instruction to be called an algorithm. These properties include:
  - a. An algorithm ends after a fixed number of steps
  - b. Each step in an algorithm clearly specifies the action to be performed
  - c. The steps in an algorithm specify basic operations.
  - d. The algorithm should accept input data, in a defined format, before it can be processed by the given instructions
  - e. An algorithm generates one or more outputs after the input is processed.
4. Algorithm for solving a particular problem can be different depending on how individuals feel such problem can be solved provided the algorithms contain all the basic properties of a good algorithm.
5. Pseudo-code represents an algorithm in the English language

## 6.0 Tutor-Marked Assignments

---

- i. Define an algorithm
- ii. List five properties of an algorithm
- iii. State the various forms of representing algorithms
- iv. Write an algorithm to compute the square root of a number

## 7.0 References/Further Readings

---

AnanyLevitin (2007) Introduction-to-the-Design-Analysis-of-Algorithms: 2nd-Edition, Pearson Addison Wesley.

I. S. Shehu "CPT 223 Lecture note (2014): Introduction to Algorithms"

# Unit 3

---

## Flowcharts

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Definition of Flowchart
  - 3.2 Flowchart Symbols and Their Meanings
  - 3.3 Applications of Flowchart
  - 3.4 Relationship between Pseudo-Code and Flowcharts
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Before going into writing computer programs to solve a problem, it is highly necessary for the programmer to do an initial design by using proper tools for such a design. To avoid unnecessary errors in the programming, the programs should be expressed in an understandable form. Generally, programmers and analysts make use of charts and shapes as very important tools. In this unit, you will be introduced to the use of flowcharts for writing programming languages.

## 2.0 Learning Outcomes

---

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

- i. Define flowchart
- ii. Identify various symbols used in flowcharts.
- iii. Construct simple flowcharts for simple programs

## 3.0 Learning Contents

---

### 3.1 Definition of Flowchart


---

Now let us discuss flowchart, flowchart can be defined as a pictorial representation to solve a problem. In other words, it is a graphical representation of an algorithm for solving a problem. As you learnt in the above unit that for you to write a program to solve a problem, you need to express the steps of solving such a problem step-by-step called an algorithm. However, flowchart is used to represent the algorithm in which the steps of processes are shown in various kinds of boxes that are orderly connected with arrows. Flowchart makes it easy for you to understand the process of solving a problem. It makes use of standard symbols use to represent different processes. The most common symbols as approved by the American National Standards Institute (ANSI) are as shown in table 3.1 of section 3.2.

### 3.2 Flowchart Symbols and Their Meanings

---

The most common symbols used in flowcharting are shown in table 3.1

<b>SYMBOLS</b>	<b>NAME</b>	<b>MEANING</b>
	Oval	Terminal



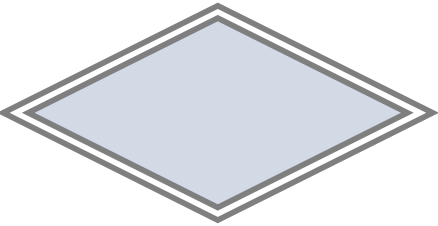
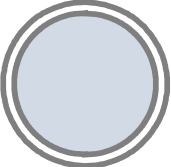


	Rectangle	Process
	Parallelogram	Input/output
	Diamond	Decision
	Circle	Connector
	Arrow	Flow arrow
		Display

Table 3.1: common flowchart symbols and meaning

**Definition of Symbols**

**Terminal:** This is used to indicate start and end of the chart

**Process:** This is the block that contain computational procedures and analyses

**Input/output:** This is used when input is read in and output is produced

**Decision:** This is the block where conditions that result in yes or no are put in

**Connector:** This is used to join segments of flowchart or where multiple arrow flows converge in a single point

**Arrow Flow:** This is used to indicate the sequence in which the steps described by the flowchart should be carried out.

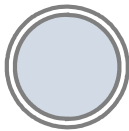
**Display:** This is used to show the output of an algorithm

### Self-Assessment Exercise(s) 1

1. Define flowchart
2. List five common symbols used in flowcharting
3. The symbol below is used for .....



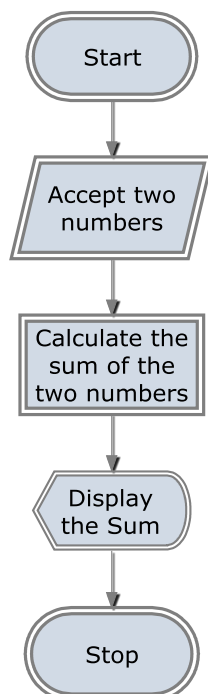
4. What is the function of the symbol below?



### 3.3 Applications of Flowchart

This section will introduce you to some simple examples of solving problems using flowcharts approach. You will first learn how to sum two numbers in a flowchart.

#### Flowchart for computing sum of two numbers



**Figure 3.1** Flowchart that displays sum of two numbers

### Flowchart to find the greater of two numbers

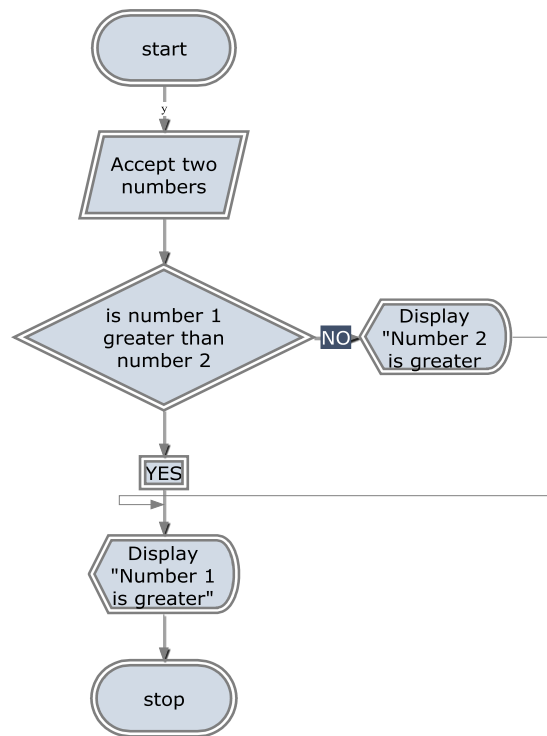


Figure 3.2: Flowchart to display the greater of two numbers

### Flowchart for computing the average of two numbers

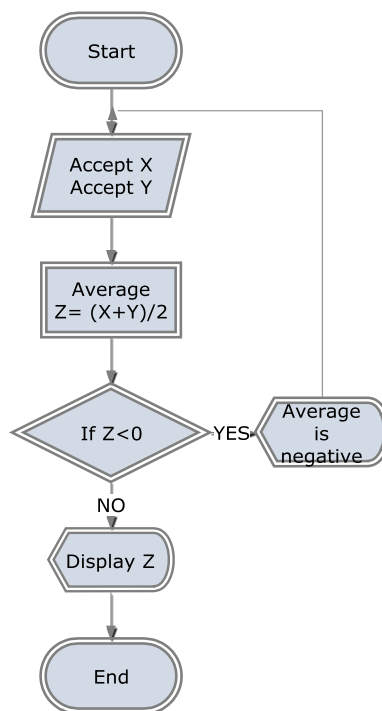

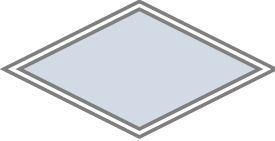



Figure 3.3: Flowchart that displays the average of two numbers

### 3.4 Relationship between Pseudo-Code and Flowchart

---

The following table shows the pseudo-code keywords with corresponding flowchart symbols

Pseudo-code Keywords	Flowchart Symbols
Accept	
If ..... else	
Begin .... End	

**Table 3.2:** Pseudo-code keywords with corresponding flowchart symbols

#### Self-Assessment Exercise(s) 2

1. The accept keyword in a pseudo-code is similar to the ..... symbol in flowchart
2. Write a pseudo-code for the flowchart shown in figure 3.2
3. If.... else in pseudo-code is similar to the ..... symbol in flowchart
4. .... symbol is used to show the outcome of an algorithm

### 4.0 Conclusion

---

This unit has introduced you to the concept of flowchart as a graphical way of representing an algorithm for solving a problem.

Some of the common symbols used for flowcharting were identified and applied to solve simple problems.

It was obvious that representing algorithms in flowcharts makes it easier to understand the processes of solving a problem.

Some pseudo-code keywords were also compared with the corresponding flowchart symbols.

## 5.0 Summary

---

Flowchart is a pictorial representation of an algorithm for solving a problem

Flowchart makes use of standard symbols

Some of the symbols and functions are:

1. **Terminal:** This is used to indicate start and end of the chart
2. **Process:** This is the block that contain computational procedures and analyses
3. **Input/output:** This is used when input is read in and output is produced
4. **Decision:** This is the block where conditions that result in yes or no are put in
5. **Connector:** This is used to join segments of flowchart or where multiple arrow flows converge in a single point
6. **Arrow Flow:** This is used to indicate the sequence in which the steps described by the flowchart should be carried out.
7. **Display:** This is used to show the output of an algorithm

## 6.0 Tutor-Marked Assignments

---

- i. Define flowchart
- ii. Show a flowchart that computes square root of a number
- iii. Differentiate between display symbol and decision symbol used in flowchart
- iv. Reproduce figure 3.2 using manual input and print output
- v. Write a pseudo-code for the flowchart in figure 3.3

## 7.0 References/Further Readings

---

J. Glenn Brookshear “*Computer Science: An Overview*” Ninth Edition (2007)

NIIT Student Guide: Fundamentals of Programming, Mastermind Series



# Module 6

---

## The Internet

Unit 1: Introduction to the Internet

Unit 2: Explain the concept of Web Technology

# Unit 1

---

## Introduction to the Internet

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 What Is a Network?
  - 3.2 Types of Networks.
  - 3.3 Importance of Networking.
  - 3.4 What Is the Internet?
  - 3.5 History of Networking and the Internet.
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings

## 1.0 Introduction

---

Imagine being the desk officer of FUT CODEL and you need to monitor the activities of other representative of the unit in various part of the country so as to know the details of the number of people who came to inquire about the activities of the unit, those who show interest in one activity or the other and the number of people who have officially registered as students in a program undertaken by the unit, by the various representatives of the unit. A possible solution is for the desk officer to visit the various representatives of the units at their various locations so as to obtain that information which may be tasking, tedious and time consuming. An alternative solution is to connect the various representatives' computers at different locations together through a network which allows them to communicate, share resources and information. This will facilitate you (the desk officer) to have access to all that information from your seat enhancing your productivity. Wow! Imagine how great that would be. You are worried that what is a network and how can it be established. You will find out in a short while.

## 2.0. Learning Outcomes

---

This unit introduces you to the fundamental concept of networking and the internet. It also traces the background and history of networking and the internet. At the end of this unit, you will be able to:

- i. explain what computer networks are, and classify them based on their types;
- ii. explain and establish the importance of network and internet;
- iii. Trace the background and history of computer networks and internet.

## 3.0 Learning Contents

---

### 3.1 What is a Network?

---

To have proper understanding of this topic, you need to know what network is;

A network is a group of computers connected together--wired or wireless--by some special protocols so that they can communicate, share resources and enhance productivity. In a typical network, there are usually one or more dedicated computers that work as servers in storing, processing and routing data/information to other computers in the network. I.e. it provides services to other computers or clients in the network. Figure 1 illustrates the structure and interconnection of computers to form a network. It comprises of three computers named computer A, B and C respectively which are connected together via wired to a central server so as to communicate, exchange file, video, voice and data and also shared resources like the central copier and printer which any computer in the network can send documents to print.

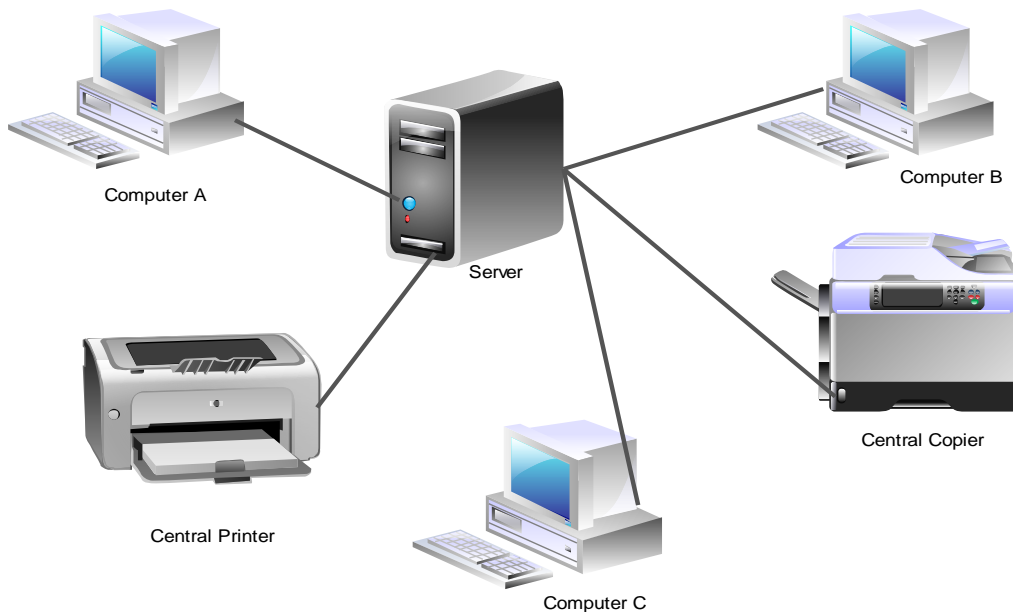


Figure 1:

An Illustration of a Typical Network

### Self-Assessment Exercise(s) 1

1. A network can be described as a group of computers connected together via a. \_\_\_\_\_ or b. \_\_\_\_\_ by some special c. \_\_\_\_\_ so that they can communicate, shared resources and enhances productivity.
2. A server is a dedicated computer used for a. \_\_\_\_\_, b. \_\_\_\_\_ and routing data/ information to other computer in the network

## 3.2 Types of Networks

Networks can be classified into four (4) main types based on their geographical coverage areas, equipment installation cost, as well as support tools and cost. The four types of networks are:

- i. **Local Area Network:** this is a network that comprises of different computers, printers, scanners and other devices connected together in a single building to enable sharing of resources, files voice and videos. Example of such network is the small network of computers, printers and other resources found in the second floor of Old SICT building of FUT, Minna, Gidan-Kwano campus which houses the head office of the CODEL Unit. It is usually abbreviated as LAN and required relatively inexpensive devices and equipment to set up and maintain
- ii. **Campus Area Network:** this is a network that connects multiple local area networks. This type of network is larger than the LANs but smaller than the metropolitan area network. Example of such network is the type of network found in a university (FUT, Minna) which connects many LANs found in various schools and units together. It is abbreviated as CANs and requires more sophisticated equipment and cost to set up and maintain when compared to LAN.

- iii. **Metropolitan Area Network:** this is abbreviated as MAN, it is a high speed network that is larger than CAN and connects many LANs in a large and densely populated urban area to the internet.
- iv. **Wide Area Network:** this is a type of network that connects many LANs over very large group of cities through the internet or over the public switched telephone network. The MANs and WANs required more sophisticated networking equipment and cost to set-up and maintain. In addition, it requires on-site administrator and support staff.

Having discussed what a network is and the different types of network, I know you kept wondering that what are the importance or benefits of this so called networks and what an internet means. You need not worry, sit back and continue to read through this unit as you will find out soon yourself.

### Self-Assessment Exercise(s) 2

1. Networks can be classified based on their a. \_\_\_\_\_ and b. \_\_\_\_\_ installation cost.
2. A network that comprises of different computers, printers, scanners and other devices connected together in a single building to enable sharing of resources, files voice and videos is known as \_\_\_\_\_ type of network.
3. What is the full meaning of the acronym MAN as related to types of network?

### 3.3 Importance of Networking

---

- i. Facilitate data resource sharing: with the advent of computer networks, it makes information sharing between computers users on the network easy. Hence, making working together easy for network users and enhancing personal and collective productivity.
- ii. Enable users to share hardware resources: networking computers enables users to share hardware resources such as printers, copier and scanners. All users within the network can access these resources.
- iii. Sharing a high-speed Internet connection: all computers connected in a network can share high speed internet in communicating with other computers within the network and also connect to other outside networks via the internet.
- iv. Networks serve as a communication and support tools: computer networks serve as a communication tool in exchanging files and other information via the network. An example of such is the electronic mail (e-mail)

### Self-Assessment Exercise(s) 3

1. Networks facilitate data resource sharing (True/False)
2. Sharing a high speed internet connection can be classified as importance of networks (Yes/No)

### 3.4 What is the Internet?

---

Be informed that internet can simply be defined as a network of networks using a common set of rules that govern the transfer of information and communication over the network known as Internet protocol (IP). The internet give you access to numerous information, files, voice and videos all over the world. A good illustration of the importance of the internet is you accessing this course material titled "Introduction to Computer" written for CODEL online irrespective of your location or receiving lecture online just by logging in to the internet and accessing the unit website wherein you then log in with your username and password.

#### Self-Assessment Exercise(s) 4

- 1..... are the set of rules that govern the transfer of information and communication over the network?
- 2.internet can be described as a network of networks (True/False)

### 3.5 History of Networking and the Internet

---

Do you know that the origin of networking concept was dated as far back as 1962, when the concept of galactic network was proposed for social interaction by J.C.R Licklider of MIT who was also the first head of research computer program at Defense Advanced Research Projects Agency (DARPA), though then known as the Advanced Research Projects Agency (ARPA). The first major landmark towards networking computer was done in 1965 when computers were made to talk together by connecting a TX-2 computer in Mass to Q-32 in California which marks the first ever wide area network built using a telephone line but with a low speed dial-up.

He proceeded to further explore this network concept by proposing the ARPANET in 1967 which, theoretically, connections would not be lost even in the failure or absence of one or more of the network components. In 1969, ARPANET has metamorphosed as a network of four computer making it the first ever internet built by the US department of defense. However, ARPANET first public demonstration took place at international conference on computer communications in 1972.

The US internet supported about 11,000 servers in 1987 when they joined their Canadian counterpart. This figure increased drastically to about 200,000 servers by the end of 1989. Most countries' individual networks were linked together to form one worldwide network of networks by early 1990s and over 1million servers were on the internet as at the end of the year 1992. As of today no one can predict the exact figure of how many computers and network users are on the internet that reflects how large the internet has become. However, it was estimated that over one (1) billion users and 500 million computers are on the internet. The internet today serves as the largest avenue for information sharing, business, entertainment and social networking.

#### Self-Assessment Exercise(s) 5

1. The concept of galactic network was proposed for social interaction by .....?
2. The first major landmark towards networking computer was achieved in ... year?
3. What is the full meaning of the acronym DARPA.....?
4. The internet today serves as the largest avenue for information sharing, business, entertainment and social networking (True/False)

## 4.0 Conclusion

---

In this unit you learnt what computer networks and internet is all about, and how to classify them based on their types. You have also learnt the importance of network and internet.

## 5.0 Summary

---

A network is a group of computers connected together via wired or wireless connection by some special protocol so that they can communicate, share resources and enhance productivity.

We have four (4) types of networks namely the Local Area Network (LAN), the Campus Area Network (CAN), Metropolitan Area Network (MAN) and the Wide Area Network (WAN) which are usually classified based on their coverage area, cost of equipment and maintenance as well as the support and administrative tools required.

Networking is essential because it facilitates the sharing of resources such as data, hardware, high speed internet connection as well as serving as communication and support tools among users within and outside the network via internet which is a network of networks.

The history of internet and networking is traced back to 1969 when the U.S. Department of Defense proposed ARPANET which connects and network four computers together. This continued to undergo changes in technology and modifications to what we have as internet of today which has over 1 billion users and 500 million computers all over the world. The internet serves as the largest avenue today for information sharing, business, entertainment and social networking.

## 6.0 Tutor Marked Assignment

---

- i. What do you understand by the term networking?
- ii. Explain the types of Networks you know.
- iii. With the aid of a suitable diagram, illustrate the concept of a network.
- iv. Define internet.
- v. Briefly highlight the importance of networks and internet
- vi. Succinctly explain the history and background of computer networks and the internet.

## 7.0 References for Further Reading

---

Computer Networking, A Top-Down Approach 6<sup>th</sup> Edition by Kurose and Ross 2006.

Technology Backgrounder on Data Networking Basics by Larry Hettick and Steven Taylor 2004

Barry M. Leiner, Robert E. Kahn, Jon Postel, Vinton G. Cerf, Leonard Kleinrock, Larry G. Roberts, David D. Clark, Daniel C. Lynch and Stephen Wolff "A Brief History of the Internet" ACM SIGCOMM Computer Communication Review, Volume 39, Number 5, October 2009

# Unit 2

---

## Explain the Concept of Web Technology

### Contents

- 1.0 Introduction
- 2.0 Learning Outcomes
- 3.0 Learning Contents
  - 3.1 Concept of Web Technologies
  - 3.2 Hyper Text Markup Language (HTML) Protocol
  - 3.3 Format of A Web Page
  - 3.4 Support Tools for Website Creation.
- 4.0 Conclusion
- 5.0 Summary
- 6.0 Tutor-Marked Assignment (TMA)
- 7.0 References/Further Readings



## 1.0 Introduction

---

Isn't it pretty cool? Having access to all kinds of information, be it data, voice, video and so on, at the snap of your fingers, via the internet. Imagine how tasking it would be for you if you have to come down all the way to CODEL at FUT, Minna to receive your lecture weekly irrespective of your location, or your daily schedules. Its hectic, right!? Thank God for the internet that is a network of networks that helps interconnect many computers around the World together. This averts the stress of having to travel tens or hundreds of kilometer down to received lecture at the main office. This was made possible by the interconnections of computers and other devices in such a way that they can communicate and share resources thereby forming what is known as a web through which you can access all lecture notes via the internet. I know your worry is "how would that be possible?" How the computer at CODEL office would know that I am requesting for a particular study guide, how would it transmit the study guide over the network to my computer? How would my computer retrieve it over the network right? Relax you will find out soon and lots more at the end of this unit.

## 2.0 Learning Outcomes

---

This unit introduces you to web technologies, the Hyper Text Markup Language (HTML) protocol, format of a web page and support tools for website creation. At the end of this unit, you will be able to:

- i. Discuss the concept of web technologies;
- ii. Explain and establish the importance of HTML protocol in web technology;
- iii. Enumerate the format of a web page; and
- iv. Highlight the various support tools for website creation.

## 3.0 Learning Contents

---

### 3.1 Concept of Web Technologies

---

Let us introduce to ourselves web technology which can simply be described as the mechanism of interconnecting computers and other computer devices in such a way that enable it to share resources and communicate together over a network. This mechanism allows message to be send from a source to a receiver and facilitating the receiver to be able to decipher the message received and send feedback or acknowledgement back to the sender. In viewing internet resources, you need a browser which enables you to read or access information on the internet network. There are several types of browsers among which include the Internet Explorer, Google Chrome, Mozilla Firefox, Opera, and Netscape. This browser hooks-up to the internet via Modem or Internet Service Providers (ISPs), to a machine remotely requesting for documents and formatting such document retrieve to a format that can be view on the internet. In achieving this, the browser used the hypertext transfer protocols (HTTP) which is a special language designed for that purpose. It is a connection oriented network in the sense that an HTTP server is being run by the remote machine that contains the document. When a request is received by the server, it processes it and sends it back to the computer (sender) for viewing via the browser. The browser locates the corresponding server to send a request to, using the associated uniform resource locator (URL) which help locate each document available on the web. The web document can take any form but the

universally acceptable standard is the hypertext markup language (HTML) as discuss in the next section.

### Self-Assessment Exercise(s) 1

1. .... described the mechanism of interconnecting computers and other computer devices in such a way that enable it to share resources and communicate together over a network.
2. A browser enables you to read or access information on the internet network (True/False).
3. .... and .... are examples of a browser?
4. What is the acronym for Hyper Text Transfer Protocols.....

## 3.2 Hyper Text Markup Language (HTML) Protocol

---

The first web page was created in November 1990 with little or no acceptable standards to follow. A group known as the World Wide Web Consortium (W3C) was then formed who were shouldered with the responsibility of developing a standard to be used for HTML in order to ensure uniformity of the protocol around the World. HTML is a language that is generally acceptable for encoding World Wide Web documents. It enables the browser to interpret and display the content of the documents retrieved from the web browsers. Furthermore, the HTML can be used in creating formatted text which remains in the same format when retrieved by a computer. HTML is made up of texts which show the content of the web page and tags that show the visual aspect and layout of the webpage. The most widely acceptable recent version of the HTML version is the HTML 4.0 which comprise of a cascading style sheets that enable repeated style features to be specify using an HTML Tags. Basically, you need a simple-text editor such as Notepad in coding your HTML so as to enable you make your own web page just like the one you view when you log on to CODEL Website.

### Self-Assessment Exercise(s) 2

1. The first web page was created in November 1990 with little or no acceptable standards to follow (True/False).
2. What is the full meaning of the acronym W3C as related to web technology.....?
3. .... is a language that is generally acceptable for encoding World Wide Web documents

## 3.3 Format of a Web Page

---

Having discussed the concept of HTML in section 2.2.2, this section will focus more on the structure of the HTML web page. All HTML page begins with a tag known as start tag and end with an end tag of html element respectively. Usually an angle bracket is used to enclose the HTML element “<>” these tags enable the computers to know that the document requested or being view is an HTML document. All other elements is then fit inside the start and end html tags.

The webpage can be classified further into two main classes namely, the Head and Body respectively as illustrated below in the HTML web page code showing My first CPT 111 CODEL online course webpage with an explanation of each line of codes done in a wall bracket. The Head contains instructions on how the web page will run its title and category while the body component of the HTML webpage contain all that will be displayed by the web browser when logging on to the webpage after creation. All visible texts are placed outside any angle brackets

You can type in the html code shown in figure (2) in a notepad, then save it using all files option format. Having done that attempt to view your web page by opening a web browser and go to file, open and select browse so as to enable you navigate to the location you saved your file. When you find the file double click it... Wow! You have just viewed your first web page.

```
<html>    (html start tag)
<head>    (head start tag)
</head>   (head end tag)

<body>    (body start tag)

<h2>My first CPT 111 CODEL online course
webpage!</h2>

      (what will display on the web page)

</body>   (body end tag)
</html>   (html end tag)
```

### Self-Assessment Exercise(s) 3

1. All HTML page begins with a tag known as a..... and end with a b. ....of html element respectively.
2. .... is used to enclose the HTML element

### 3.4 Support Tools for Website Creation

---

To have better understanding there are numerous support tools for website creation most of which are available freely online. Some of these tools will be highlighted as detailed explanation on how to use them in creating your web page will be discussed in other courses at higher level.

#### Tools for Website Creation

- i. **BugHerd:** this is a tool that is used by web developer in handling and organizing feedback, fixes bug as well as feature requests.
- ii. **Fontello:** it helps in customizing the web page fonts

- iii. **Proto.io:** this is very useful in including animations to your webpage and also enables sharing such animations and commenting
- iv. **Foundation 3:** this serve as a template that can be used by web developer when creating their own web page
- v. Brackets assists web developer to cater for repetitive tasks such as searching a function, reloading a browser and so on, incorporated into the development process.

Other website creation tools include Dreamweaver CS6, Cloud9 IDE, Adobe Edge Inspect, Trello, Gridset, Microsoft WebMatrix 2, Firefox 18, Blogger, Google Sites, Jimdo, Kafafa, Wikispaces and so on

#### Self-Assessment Exercise 4

- i. List any three Support Tools for Website Creation

## 4.0 Conclusion

---

In this Unit, you have learnt about the concept of Web Technologies, the importance of HTML Protocol in Web Technology, the Format of a Web Page and the also the various support tools for Web Creation.

## 5.0 Summary

---

I believe we understand that web technology is the mechanism of interconnecting computers and other computer devices in such a way that enable it to share resources and communicate together over a network.

A browser enables you to read or access information on the internet network. There are various types of browsers among which include the Internet Explorer, Google Chrome, Mozilla Firefox, Opera, and Netscape.

Hyper Text Transfer Protocols (HTTP) is a special language or set of rules designed to format all messages, documents and etc. on the internet so as to be view using the web browser.

HTML is a language that is generally acceptable for encoding World Wide Web documents. A sample HTML web page was created to give you a feel of how to create your own web page. In addition, several support tools for website creation such Bugherd, Fontello, Foundation 3, Firefox 18, Blogger, Google Sites and so on.

## 6.0 Tutor Marked Assignment

---

- i. Define the term Web Technology
- ii. What did you understand by the word browser?
- iii. List any five types of browser you know
- iv. Explain how the browser hooks up to the internet in accessing documents
- v. What is the full meaning of URL?
- vi. HTML stands for .....

- vii. Which body standardized the HTML?
- viii. Explain what you understand by the term Hyper Text Markup Language Protocol
- ix. State one simple-text editor require for HTML web page creation
- x. Define HTML Tags
- xi. Briefly explain any five support tool required for website creation

## 7.0 References for Further Reading

---

Computer Networking, A Top-Down Approach 6<sup>th</sup> Edition by Kurose and Ross 2006.

Duyne, Douglas K. Van, James Landay, and Jason I. Hong. The design of sites: patterns, principles, and processes for crafting a customer-centered Web experience. Addison-Wesley Longman Publishing Co., Inc., 2002.

Bonk, Curtis J. The world is open: How web technology is revolutionizing education. John Wiley & Sons, 2009.

Hodgkins, R. (n.d.). Understanding web technology. Retrieved June 3, 2015, from <http://www.computerweekly.com/feature/Understanding-web-technology>

## Answers to Self-Assessment Exercises

---

### MODULE 1

#### UNIT 1

##### SELF-ASSESSMENT EXERCISE 1:

- i. A computer is an electronic device that processes data, in order to convert it to information that is useful to people
- ii. True
- iii. False
- iv. Accepting data, processing data to get information, displaying the information, as well as storing it.

##### SELF-ASSESSMENT EXERCISE 2:

- i. Vacuum tubes
- ii. False
- iii. True
- iv. Integrated circuits (ICs)
- v. Microprocessor

#### UNIT 2

##### SELF-ASSESSMENT EXERCISE 1:

- i. CPU, memory, input devices, output device, and storage
- ii. storage
- iii. false
- iv. memory

##### SELF-ASSESSMENT EXERCISE 2:

- i. Similarity: Both RAM and storage can be written to and read from. Difference: RAM is volatile, while storage is non-volatile.
- ii. Similarity: Both ROM and storage are non-volatile. Difference: storage can be written to many times, whereas ROM is not usually written to multiple times.
- iii. Flash drives are very popular because they are portable and can be used on any computers having USB ports. Furthermore, they do not require any burners or devices to operate them.

### MODULE 2

#### UNIT 1

##### SELF-ASSESSMENT EXERCISE 1:

1. system
2. application
  - a. compilers
  - b. debuggers

## UNIT 2

### SELF-ASSESSMENT EXERCISE 1:

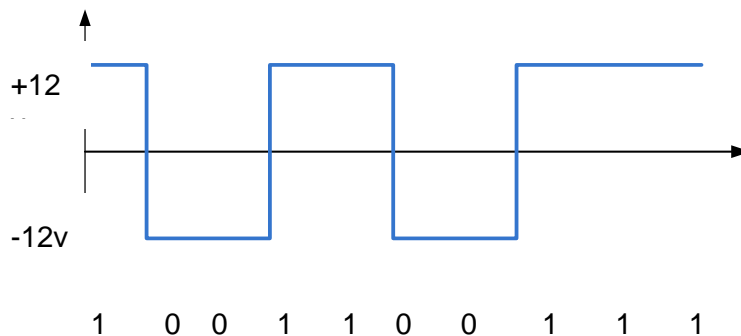
- i. Serial processing
- ii. Monitor
- iii. Because I/O devices are slow compared to the processor
- iv. Mainframe operating systems, Server Operating Systems, Personal Computer Operating Systems, Handheld Computer Operating Systems, etc.

## MODULE 3

### UNIT 1

#### SELF-ASSESSMENT EXERCISE 1

1. 2 states
2. 8 bits
3. Each bit is 1 second, as entire pulse lasts for 10 seconds comprising of 10 bits



#### SELF-ASSESSMENT EXERCISE 2

1. 8388508
2. 512 files
3. 0.078 seconds

#### SELF-ASSESSMENT EXERCISE 3

1. a) 1000 b) 0010 c) 1011 remainder 11 d) 1001111100
2. Addition : 1111011 Subtraction : 1100001 multiplication: 10110010110 division: 1000 remainder 110

## UNIT 2

#### SELF-ASSESSMENT EXERCISE 1

1. 174
2. 177

#### SELF-ASSESSMENT EXERCISE 2

1. 100110011
2. 100101000

#### SELF-ASSESSMENT EXERCISE 3

1. a) 6BDA b) FFFFAA5517 c) 69

2. a) 0111101011011000      b) 0111000110111101      c) 000100001010

SELF-ASSESSMENT EXERCISE 4

1. a) 100010      b) 010010      c) 10011000  
 2. a) 6            b) 6            c) 8  
 3.

	Excess Notation	Integer Represented
	11111	15
	11110	14
	11101	13
	11100	12
	11011	11
	11010	10
Positive integers	11001	9
	11000	8
	10111	7
	10110	6
	10101	5
	10100	4
	10011	3
	10010	2
	10001	1
Change of sign	10000	0
	01111	-1
	01110	-2
	01101	-3
	01100	-4
	01011	-5
	01010	-6
Negative integers	01001	-7
	01000	-8
	00111	-9
	00110	-10
	00101	-11
	00100	-12
	00011	-13
	00010	-14
	00001	-15
	00000	-16



## SELF-ASSESSMENT EXERCISE 5

1. a) binary number = -101.0001, decimal number = -5.0625  
b) binary number = 111.1010, decimal number = 7.3125

## UNIT 3

### SELF-ASSESSMENT EXERCISE 1

1. 100011010101011010100010110010011011111000
2. Hello.

## MODULE 4

### UNIT 1

SELF-ASSESSMENT EXERCISE 1: i) True ii) Logic Gate iii) False

SELF-ASSESSMENT EXERCISE 2: i) Logic Gate ii) AND, OR, NOT operation/operators.  
iii) True

### UNIT 2

SELF-ASSESSMENT 1: i) Memory. ii) Secondary Memory iii) False iv) True v) True

SELF-ASSESSMENT 2: i) Cache Memory ii) True

### UNIT 3

SELF-ASSESSMENT 1: i) John von Neumann. ii) True iii) False. iv) Fetch, Decode and Execute

## MODULE 5

### UNIT 1

#### SELF-ASSESSMENT EXERCISE 1:

1. Machine Language, Assembly Language and High-Level Language
2. Machine Language
3. Assembly language is a machine-dependent language

#### SELF-ASSESSMENT EXERCISE 2:

1. A compiler
2. Language Specific
3. Translator

### UNIT 2

#### SELF-ASSESSMENT EXERCISE 1:

1. Algorithm
2. Pseudo-code and flowchart
3. Program

## SELF-ASSESSMENT EXERCISE 2:

1.
  - Algorithm ends after a fixed number of steps
  - Each step in an algorithm clearly specifies the action to be performed
  - An algorithm generates one or more outputs after the input is processed
2. Pseudo-code represents an algorithm in the English language to solve a problem.
3. Pseudo-code to find the mean of a set of numbers is as shown below:  
**Input:** n numbers items  
**Output:** The arithmetic mean of the numbers.

The steps involved in calculating the mean of the n set of numbers are as follows:

1. n = read input from user for list of n
2. Sum = 0
3. i = 0
4. While  $i < n$  {
5. Number = read input from user
6. Sum = sum + number
7.  $i = i + 1$ }
8. Mean = sum / n

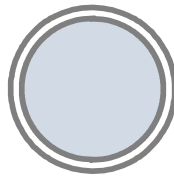
## UNIT 3

### SELF-ASSESSMENT EXERCISE 1:

1. Flowchart is a pictorial representation of an algorithm for solving a problem.
2. The following is a list of symbols commonly used in flowcharting:  
Terminal: This is used to indicate start and end of the chart  
Process: This is the block that contain computational procedures and analyses  
Input/Output: This is used when input is read in and output is produced  
Decision: This is the block where conditions that result in yes or no are put in  
Connector: This is used to join segments of flowchart or where multiple arrow flows converge in a single point  
Arrow Flow: This is used to indicate the sequence in which the steps described by the flowchart should be carried out.  
Display: This is used to show the output of an algorithm
3. The symbol below is used to show the *output of an algorithm*



4. **Connector:** This is used to join segments of flowchart or where multiple arrow flows converge in a single point



**SELF-ASSESSMENT EXERCISE 2:**

1. The accept keyword in a pseudo-code is similar to the Parallelogram Symbol in flowchart

2. The pseudo-code for figure 3.2 is as shown below:

begin:

Accept the first number

Accept the second number

If the first number is greater than the second number

display the “the first number is greater”

else

display “the second number is greater”

end

3. If.... else in pseudo-code is similar to the diamond symbol in flowchart
4. Display is used to show the outcome of an algorithm

**MODULE 6**

**UNIT 1**

**SELF-ASSESSMENT EXERCISE 1:**

- i. a. Wired, b. Wireless, c. Protocols
- ii. a. Storing, b. Processing

**SELF-ASSESSMENT EXERCISE 2:**

- i. a. Geographical Coverage Areas, b. Equipment
- ii. Local Area Network (LAN)
- iii. Metropolitan Area Network

**SELF-ASSESSMENT EXERCISE 3:**

- i. True
- ii. Yes

**SELF-ASSESSMENT EXERCISE 4:**

- i. Internet Protocol
- ii. True

**SELF-ASSESSMENT EXERCISE 5:**

- i. 1965

- ii. Defense Advanced Research Projects Agency
- iii. True

## **UNIT 2**

### **SELF-ASSESSMENT EXERCISE 1:**

- i. Web Technology
- ii. True
- iii. Internet Explorer, Google Chrome, Mozilla Firefox, Opera, and Netscape
- iv. HTTP

### **SELF-ASSESSMENT EXERCISE 2:**

- i. True
- ii. World Wide Web Consortium
- iii. Hyper Text Markup Language (HTML)

### **SELF-ASSESSMENT EXERCISE 3:**

- i. a. Start tag, b. end tag
- ii. Angle bracket "<>"

### **SELF-ASSESSMENT EXERCISE 4**

- i. Foundation 3, Dreamweaver CS6, Bugherd, Fontello, Microsoft WebMatrix 2 etc.



DAYSPRINGERS PRODUCTION  
+2347086257747