SCHEME OF WORK

FIRST TERM E-LEARNING NOTE **CLASS: JSS 2 (BASIC 8)**

SUBJECT: COMPUTER STUDIES

WEEK TOPIC

SUB-THEME: BASIC COMPUTER OPERATIONS CONCEPTS

1. **Revision of last term's work.**

2. **Computer Software:** (a) Definition of software (b) Types of examples of software: - system software (operating system).

3. **Computer Software:** (b) Examples of software cont'd; application software (word processing, spreadsheet, graphics)

4. **Operating System:** (a) Definition of operating systems (OS) (b) Examples of operating systems (DOS, Windows, Linux, Unix)

5. **Operating System:** (c) Functions of operating systems: (Resource allocation, monitoring, utilities).

6. **Units of Storage in computer:** (a) Units of storage: - Nibble Byte, Kilobyte, Megabyte, Gigabyte, etc

7. **Computer Problems Solving skills:** Computer programs; meaning, Examples (Logo, BASIC).

8. **BASIC Program:** (a) BASIC Language: Meaning of BASIC, BASIC character set. (b) Key BASIC statement (c) Simple BASIC statement

9-10. Revision

11-13 Examination

REFERENCES:

- **HiiT Plc (2016) Computer Studies for Junior Secondary Education (JS1), HiiT Publisher, Nigeria Computer Society (NCS).**
- **Basic Science and Technology: Computer Studies for Junior Secondary Schools 1 by Wole Olatokun et al**

Searchlights on Computer Studies for JSS, Bk 1 by Kayode Owolabi

WEEK 2

TOPIC: COMPUTER SOFTWARE

CONTENT

(i) Definition of Software

(ii) Types of Software

(iii) Examples of Software

Subtopic 1: Definition of Software

Software, also called a program, is a series of instructions that tells the computer what to do and how to do it. In other words, it is a collection of computer programs, procedures and documentation that perform different tasks on a computer system. The software is a program written by a programmer or software developer. Software is the intangible part of the computer system though they come in CD's according to today's standard. Without the software installed in the computer, the computer is just a junk of hardware components. Therefore, software is as important to the computer system as human soul is important to human body.

Installation: This is the process of transferring the set of instruction that makes up software into the computer so as to make the computer perform certain task.

TYPES OF SOFTWARE

1. System Software
2. application Software

System Software: this consists of the program that control or maintain the operations of the computer and its devices. It serves as the interface between the user, the application software and the computer's hardware. They are programs that enhance the performance operations of computer.

TYPES OF SYSTEM SOFTWARE

1. Operating System
2. Language Translators
3. Utilities
4. Device Driver

EVALUATION:

1. Define the term Software
2. Who develop software?
3. Mention four (4) types of software

READING ASIGNMENT

Students are to study and briefly explain the meaning of Application Software

WEEKEND ASSIGNMENT

1. MS-DOS is an acronym which stands for

(a) Microsoft DOS Opening Software (b) Microsoft Disk Opening System (c) Microsoft Disk Operating Service (d) Microsoft Disk Operating System

2. A system software include _____

(a) DBASE (b) RAM (c) Word processing (d) UNIX

3. What is installation?
4. System software can be divided into how many parts?
(a) 6 (b) 8 (c) 4 (d) 1
5. _______ is collection of program available in the computer system or set of instructions guiding the operation of the computer
(a) Software (b) Firmware (c) People ware (d) Hardware

WEEK 3

TOPIC: COMPUTER SOFTWARE

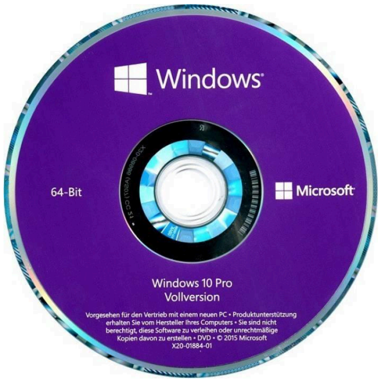
CONTENT

i. Types of System Software explained
ii. Application Software (word processing, spreadsheet, graphics)

Subtopic 1:

Types of System Software

1. **Operating system:** This is a set of program that coordinates all the activities among computer hardware devices. It also contains instructions that allow users to run application software. When a user starts a computer, portions of the operating system (OS) load into memory from the computer's hard disk. It remains in memory while the computer is on. The operating system provides a means for users to communicate with other software.



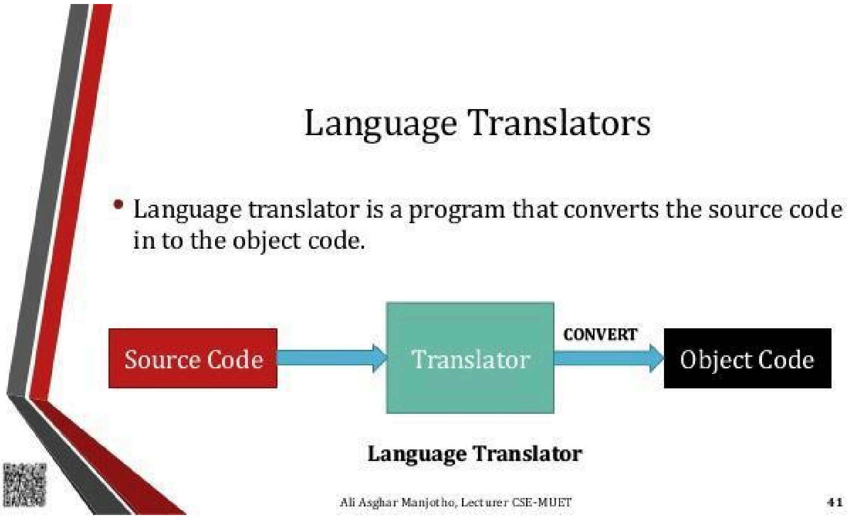
Mac
Powered By
GNU/Linux
bada
clearOS
symbian
OS
android
palm
Java
CentOS

Examples of Operating System

i. Microsoft Disk Operating System(MS-DOS)
ii. Windows OS
iii. Linux
iv. Unix
v. Xenix
vi. Ubuntu
vii. Novell Netware
viii. MAC OS

ix. Symbian OS
x. Palm OS
xi. JAVA etc.

2. **Language Translator:** The computer is like human beings that understands language and can also interact with the user. But the only language the computer understands is called machine language. This machine language is made up of only 0’s and 1’s. Humans do not use this kind of language but rather English language. When a computer program is written in English language form, it is said to be a High Level Language. Because humans must instruct the computer machine on how to carry out specific task, there is need for them to write programs that will instruct the computer machine. But humans cannot use machine language because it is very difficult to write a program in that language. Therefore, the language translator now helps to translate the program written in English language (high Level Language) into Machine Language (0’s and 1’s) for the computer to understand and then perform the task. **A program written in high-level language is called source code while the program format it is translated into is called object code (machine code). To convert the source code into object code (machine language which is made up of 0’s and 1’s), Language translators are needed.**



Language Translators

- Language translator is a program that converts the source code in to the object code.

Source Code → Translator → CONVERT → Object Code

Language Translator

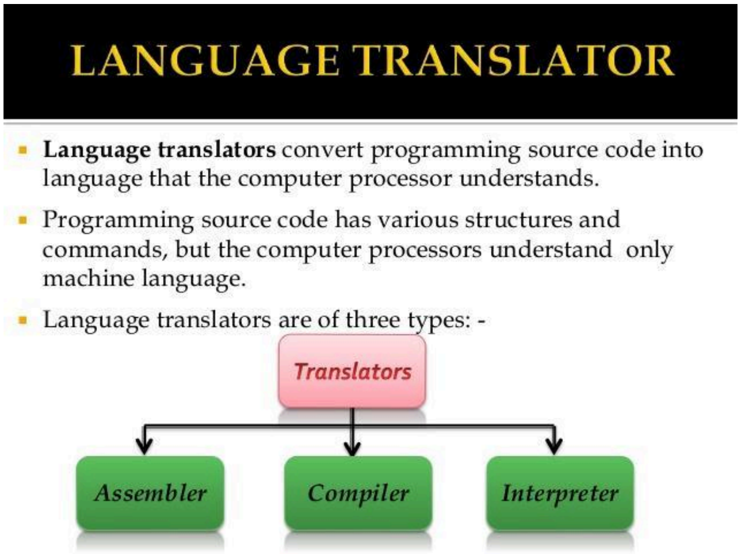
Ali Asghar Manjotho, Lecturer CSE-MUET 41

EXAMPLES OF TRANSLATOR

i. **Compiler:** Compiler is a language translator which is used to convert programs written in High-Level Language all at once to low-level language. It translates the entire program and also reports the errors in source program encountered during the translation.

ii. **Interpreter:** Interpreter is a language translator which is used to convert programs in high-level language to low-level language. Interpreter translates the program line by line and reports the error once it encountered during the translation process.

iii. **Assemblers:** Assembler is a language translator which is used to translate a program written in Assembly language to machine language code.



LANGUAGE TRANSLATOR

- **Language translators** convert programming source code into language that the computer processor understands.
- Programming source code has various structures and commands, but the computer processors understand only machine language.
- Language translators are of three types: -

Translators

Assembler

Compiler

Interpreter

3. **Utility**: this is a kind of system software designed to help analyze, configure, optimize and maintain the computer. **A single piece of utility software is usually called a utility or tool. T**hey are used for general house-keeping functions of the computer such as repairing the computer, copying, sorting files etc.

EXAMPLES OF UTILITY

i. Norton utilities
ii. Antivirus,
iii. Scandisk

iv. Windows explorer, etc

4. **Device Driver:** A device driver is a computer program that operates or controls a particular type of device that is attached to a computer. A driver provides a software interface to hardware devices, enabling operating systems and other computer programs to access hardware functions without needing to know precise details of the hardware being used.

A driver communicates with the device through the computer bus or communications subsystem to which the hardware connects. When a calling program invokes a routine in the driver, the driver issues command to the device. Once the device sends data back to the driver, the driver may invoke routines in the original calling program. Drivers are hardware dependent and operating-system-specific.

DRIVERS ARE A SET OF INSTRUCTIONS THAT INTRODUCE YOUR PC TO A HARDWARE DEVICE
YOUR PC COMMUNICATES WITH THE DEVICE
© LIUTILITIES.COM 2009

Subtopic 2:

Application Software: Application software consists of software that performs specific tasks for users.

CLASSIFICATION OF APPLICATION SOFTWARE

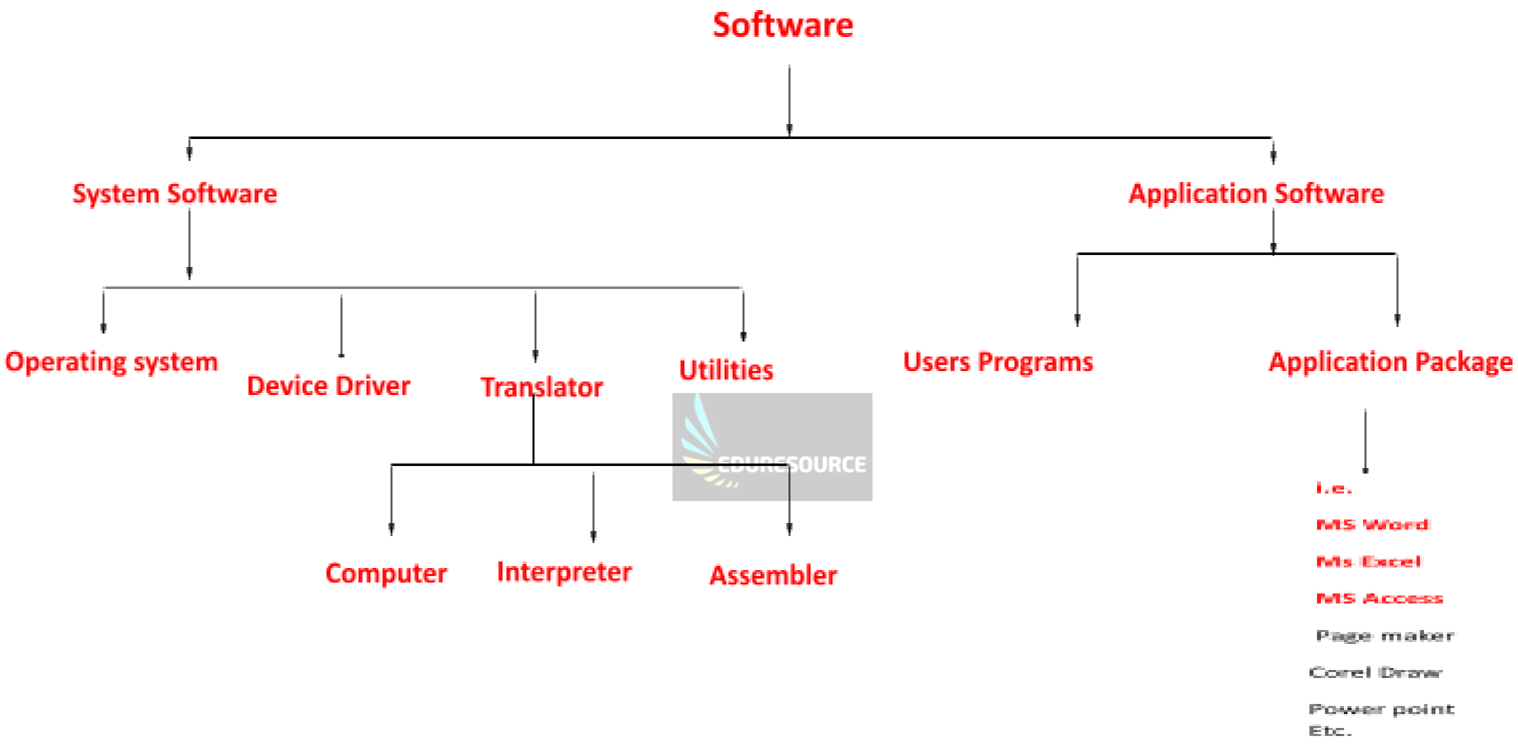
1. **User application program:**
2. **Application Packages:** This is a software package that is created for a specific purpose or industry

TYPES OF APPLICATION SOFTWARE

i. Word processing software
ii. Spreadsheet software
iii. Database management software
iv. Presentation software

v. Graphic software
vi. Communication software
vii. Multimedia software

The tree below summarizes the classes of software



EVALUATION:

1. Distinguish between system software and application software
2. List five application packages you know

READING ASSIGNMENT:

Students are to study on system and application software.

WEEKEND ASSIGNMENT:

1. A software that comes with a new computer system is called ________ software
   (a) Application (b) System (c) Functional (d) Manufacturer
2. System software can be divided into how many parts?

(b) 6 (b) 8 (c) 4 (d) 1

3. An application package that can be used to communicate textual data to the user is called ___

(a) Power point (b) Corel draw (c) MS-WORD (d) Excel

4. Data is entered into the cells of the sheet in _______ software

(a) Graphic (b) DBMS (c) Multimedia (d) Spreadsheet

5. ________ software allows users to create and play audio and video media.

(a) Presentation (b) Multimedia (c) Spreadsheet (d) Communication

6. Which of the following is application software?

(a) (a) CD-ROM (b) FORTRAN (c) ZENIX (d) SPREADSHEET

WEEK 4

TOPIC: OPERATING SYSTEM

CONTENT

1. Definition of operating system (OS)
2. **Examples of Operating system**

SUBTOPIC 1: DEFINITION OF AN OPERATING SYSTEM

DEFINITION OF AN OPERATING SYSTEM

1. **Operating system:** This is a set of program that coordinates all the activities among computer hardware devices. It also contains instructions that allow users to run application software. When a user starts a computer, portions of the operating system (OS) load into memory from the computer's hard disk. It remains in memory while the computer is on. The operating system provides a means for users to communicate with other software. It is the operating system that makes the computer understands the user's instructions and command. It is the life-wire of a computer system. It does the loading and schedule of all activities going on in a computer system.

For users to be able to interact with the computer machine, the operating system provides the user with what is called **user interface (UI)**

What is user interface (UI)? This is the space through where interactions between humans and machine occur. The goal of UI is to allow effective operation of the computer machine.

Types of User Interface

i. Command based interface

ii. Graphical User Interface (GUI): with graphical user interface, you interact with the software using text, graphics, and visual images such as icons. An icon is a small image that represents a program, an instruction, or some other object on the computer screen.

EVALUATION

1. Define the term Operating System
2. Mention what establishes a relationship between the user and the computer machine
3. A pictorial representation on the computer screen is called what?
4. Mention the two types of user interfaces

Sub-topic 2:

TYPES OF OPERATING SYSTEM

There are different types of operating system as specified by their operational performance. The most commonly used operating systems is:

1. Single user Operating System
2. Multi User Operating System
3. Distributed Operating System
4. Multiprocessing Operating System

Single user operating system

Types of single user OS:

i. **Single User/Single tasking OS:** allows only a single task performed at a time. It allows a single user to operate the machine at a time in an interactive mode and allows one user program to be in the main store and processed at a time. It does not allow users program. Examples of a single user operating system include Palm OS for Palm handheld computers.

ii. **Single User/Multitasking OS:** allow several programs to run at the same time by a single user. **Example is Window 95, Windows NT, Workstation and Windows 2000 professional.**

Multi-User Operating System: this has been designed for more than one user to access the computer at the same or different time. It permits the execution of more than one task at a time. It shares the system resources to more than one user by connecting the individual user on the system.

i. **Time sharing OS**
ii. **Batch Processing OS**

These operating systems are types of Multi-user OS in which CPU time is divided among the users. The division is made on the basis of a schedule.

Examples are UNIX, Linux.

Distributed Operating System: It manages a group of distinct computers and make them appear to be a single computer. This is where distributed applications are running on multiple computers linked together communications. A distributed operating system is an extension of the network operating system that supports higher levels of communication and interaction of the machine on the network. Distributed computations are carried out on more than one computer machine. When computers in a group work in cooperation, they form a distributed system. The computers involved in the use of this OS communicate through LAN / WAN network. **Examples of Distributed OS are internet and Intranet (the portion of the internet which is managed by an organization).**

Multiprocessing Operating System: this is the operating system that supports the use of two or more central processing unit (CPU's) within a single computer system. It also refers to the ability of an operating system to support more than one processor or the ability to allocate tasks between them. **Examples of multi processing operating system are Windows NT, 2000, XP, and UNIX.**

EVALUATION

1. State 5 examples of operating system
2. List the types of operating system

GENERAL EVALUATION

1. Explain any TWO types of operating system
2. Highlight two examples each of the types of operating system

READING ASSIGNMENT

Study and summarize the topic 'Functions of operating system'

WEEKEND ASSIGNMENT

1. Which of the following is NOT an example of an operating system?

(a) Linux (b) UNIX (c) Microsoft office (d) Windows 98

2. The following are examples of networking operating system EXCEPT

(a) Windows NT (b) MS-DOS (c) Novel Netware (d) Windows XP

3. The full meaning of Windows Me is __________

4. The latest operating system developed by Microsoft is _________

WEEK 5

TOPIC: OPERATING SYSTEM

CONTENT

Function of the Operating System

1. **Resource Allocation:** operating system allocates computer resources such as CPU time, main memory (RAM), secondary storage and input and output devices for use.
2. **File Management:** It helps in the transfer of file from one storage devices to another. It also allows all files to be easily changed and modified through the use of text editors or some other file manipulating software.
3. **Memory Management:** It allows the allocation of the main memory and other storage areas to the system programs as well as user programs and data.
4. **It Interprets commands and instructions**
5. **It coordinates and assigns compilers, assemblers, utility programs and other software to the various users of the computer system.**
6. **System Monitoring:** Operating system monitors jobs submitted for executions to ensure efficient processing.
7. **Input/output management:** it assigns different input and output device while one or more programs are being executed.
8. **It facilitates easy communication between the computer system and the computer operator.**
9. **It performs automatic transition from job to job as directed by special control statement.**
10. **It determines and maintains the order in which jobs are to be executed in the computer system.**

EVALUATION:

1. List five functions of the operating system

GENERAL EVALUATION

1. Explain any TWO types of operating system

2. Highlight two examples each of the types of operating system

3. State at least five versions of Windows Operating System you know.

READING ASSIGNMENT

Study and summarize the topic 'Units of storage in computer'

WEEKEND ASSIGNMENT

1. The functions of an operating system include the following EXCEPT

(a) allocates computer resources and time to various tasks (b) detects errors and provides error messages (c) performs memory management (d) requires special computer science skill

2. The full meaning of GUI is

(a) Ground Utility Interface (b) Graphical User Interface (c) Graphical Utility Icon (d) Global User Index

State whether the following are true or false:

3. The computer's memory resides in the computer's hard disk

4. DOS has a graphical user interface

5. Windows 7 is the latest operating system

6. Linux is a UNIX-based operating system

WEEK 6

TOPIC: UNITS OF STORAGE IN COMPUTER: Units of Storage; Bits, Nibble, Byte, Kilobyte, Gigabyte, etc.

CONTENTS:

1. Units of Storage in Computer
2. Binary Digit (Bit)

3. Nibble
4. Binary Table (Byte)
5. Word Length

Units of Storage in Computer

Computer memory is a storage location where data / information are kept or stored. All data, information, text, numbers, audio, video etc are converted into 0's and 1's and stored in different types of storage in order to be processed by the computer CPU. Hence, we have various units of storage. These are;

1. Bit

2. Nibble

3. Byte

4. Word

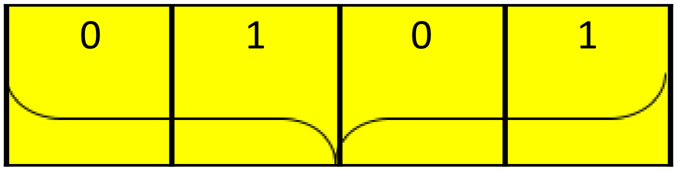
Binary Digit (Bit)

It is the first and the smallest unit of computer memory. It is a unit of data that can be either of the two conditions – **0** or **1.** Groups of bits make up storage units in the computer, called bytes (characters), words which are manipulated as a group. Each of the 0 and 1 in the diagram below indicates a bit.

0	1	0	0	1	1	0	1
bit 1	bit 2	bit 3	bit 4	bit 5	bit 6	bit 7	bit 8

Nibble

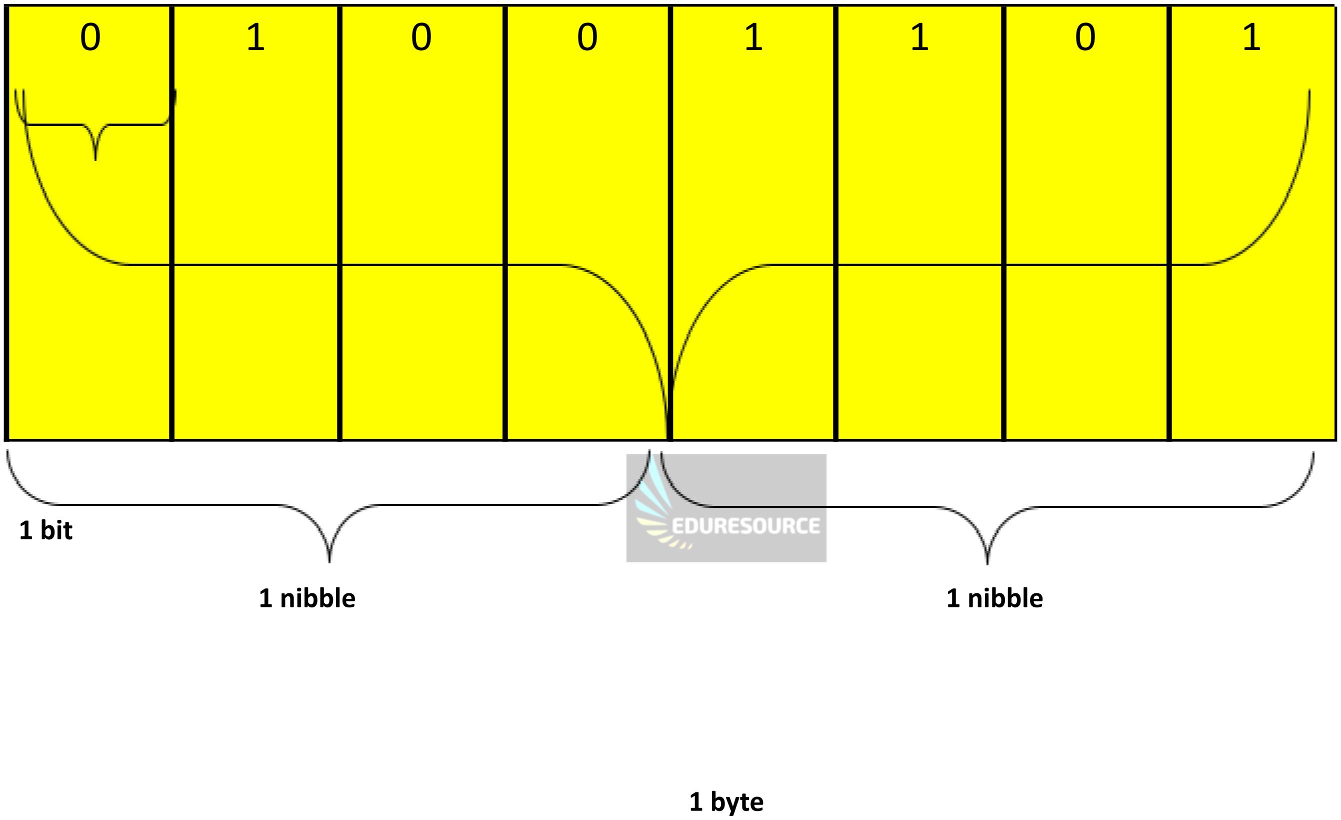
It is a storage unit that is in a sequence of 4 bits. It is equivalent to 4 bits (half of a byte) i.e 0101, 1110, 0101, 1101, 1001 etc. E.g



(1 Nibble)

Binary Table (BYTE):

A group of 8 bits is called a **byte.** A byte is the basic and the most common unit in the computer's memory. One byte is one character. A character can be a number, letter or symbol. Each byte resides temporarily on the computer's memory and this specific location is called an address. i.e. 01011101, 11110110 01010110 etc



Other higher memory units associated with bytes are:

i. Kilobyte

ii. Megabyte

iii. Gigabyte

iv. Terabyte

KILOBYTE (KB): This is equivalent to 2^{10} =1,024 bytes.

MEGABYTE (MB): A megabyte is equivalent to 2^{20} = 1,048,576 bytes

GIGABYTE (GB): A gigabyte is equivalent to 2^{30} = 1,073,741824 bytes

TERABYTE (TB): A terabyte is equivalent to 2^{40} = 1,099,511,627,780 bytes

EVALUATION

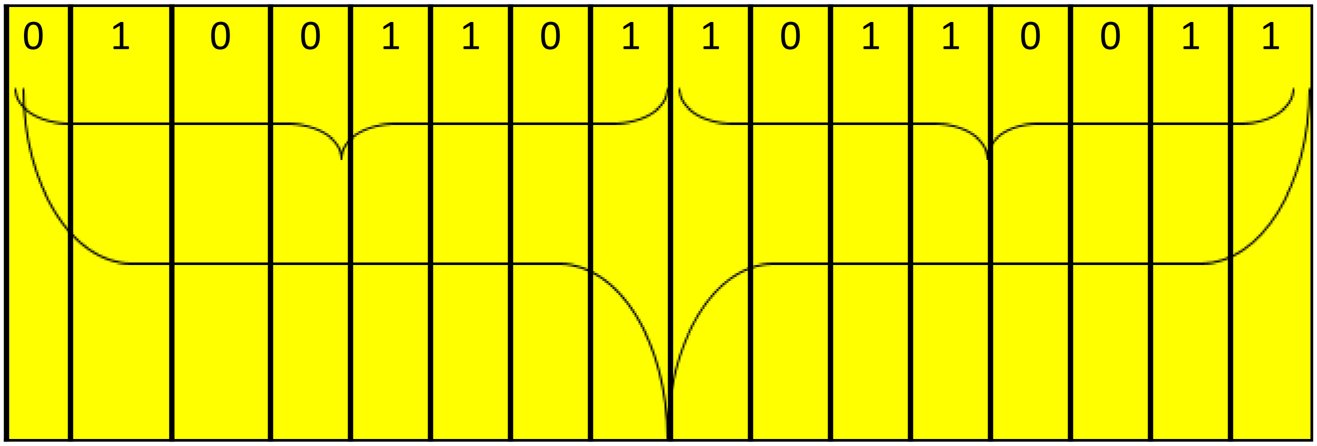
1. How many bits are in a byte?
2. What is a kilobyte?
3. List three units of storage

ASSIGNMENT

1. What is a quad word?
2. If a computer hard disk has the memory space of 3200000 bytes, express this in megabytes.
3. How many kilobytes can be found in 55000 bytes?

WORD LENGTH:

A word is a group of fixed numbers of bytes which always varies from computer to computer but is fixed for each computer. The length of a computer word is called a word length, which is in the range of 1 byte to 64 bytes. It implies the number of bits which can be handled in a single operation.



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

1 byte (8 bits) **1 byte (8 bits)**

1 word (16 bits)

Other higher memory units associated with WORD are:

i. **Double word** – is made up of 2 words which is equal to **4 bytes (32 bits)**

ii. **Quad word** – is made up of 4 words i.e. 2 double words, which equals to **8 bytes (64 bits)** etc.

WORKING EXAMPLES

1 byte = 8 bits

1 kilobyte (KB) =2^{10}= 1,024 bytes

1 megabyte (MB) = 2^{10} x2^{10} =2^{20} = 1,048,576 bytes

I gigabyte (GB) = 2^{10} x2^{10}x2^{10}= 2^{30} = 1,073,741,824 bytes

I terabyte (TB) = 2^{10}x2^{10}x2^{10}x2^{10}=2^{40} = 1,099,511,627,780 bytes

FURTHER WORKINGS

Bit = 0 or 1

4 bits = I nibble

2 nibbles = 8 bit and 8 bit = 1 byte

3 nibbles = (3 x4) bits = 12bit

4 nibbles = (4x4) bits = 16 bit

5 nibbles = (5 x4) bits = 20 bits

Conversion of bytes to bits

1 bytes = 8 bits

2 bytes = 16 bits

3 bytes = 24 bits

5 bytes = 40 bits

Unlike the metric system where I kilometer (km) is equal to 1000 meter the sub-units of byte are converted as follows.

EXAMPLE 1

Converts 2 kilobytes to byte

Solution

1 KB = 1024bytes

2 KB = 2(1024) bytes = 2048 bytes.

EXAMPLE 2

Convert 4KB to Byte

Note IKB = 1024 bytes

4KB = 4(1024) bytes

= 4096 bytes

EXAMPLE 3

Convert 2048 bytes to kilobytes

Recall,

1024 bytes = IKB

2048 bytes = 2048bytes x 1KB / 1024 bytes

=2 kilobyte

EXAMPLE 4

Change 3 megabytes to byte

1MB = 1,048, 576 bytes

3 MB = 3(1048, 576 bytes)

= 3,145,728 bytes

Convert words to bytes

Recall 1 word = group of 2 bytes

a) 1 word = 2 bytes

2 bytes = 16bits

b) 2 word = 2(2bytes)

= 2x 16bits

= 32 bits

c) 4 word = 4(2 bytes)

= 4 x 16 bits

= 64 bits

d) Change 3,145,728 Byte to MB

Solution: To change 3,145,728 to MB

1, 048,576B = 1MB

3, 145,728 = 3,145,728 x 1MB / 1,048,576

= 3MB

EVALUATION

I. Explain the following terms

(a) Nibble (b) Word (c) Double words (d) Quad word (e) Byte (f) Kilobyte (g) Megabyte (h) Gigabyte

2. Convert, (i) 3 bytes to bits (ii) 5 nibbles to bit (iii) 2 words to byte

READING ASSIGNMENT:

Read to revise this lesson again, and be prepared for questions on it

WEEKEND ASSIGNMENT:

Convert the followings: (i) 40 megabyte to kilobyte (ii) 356 kilobyte to byte (iii) 6bytes to nibble.

WEEK 7 MID TERM HOLIDAY

WEEK 8

TOPIC: COMPUTER PROBLEM SOLVING SKILL

Meaning of computer program

A computer program is a set of instructions that directs a computer to perform tasks.

Programming Language

A programming language is a set of words, symbols and codes that enables a programmer to communicate instructions to a computer.

A programmer is someone who writes and modifies computer programs.

Programming is the act of writing instructions for computer to perform a specific task.

CATEGORIES OF PROGRAMMING LANGUAGES

There are a number of programming languages existing today. Each language has its own rule for writing the instructions. Programming languages are designed for specific purposes, such as scientific applications, business solutions or web page development.

Programmers must decide which programming languages and tools to use when they write programs.

TYPES OF PROGRAMMING LANGUAGES

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

Machine Language

A language in which the computer performs the instructions immediately without any further translation is called machine language. The machine language is the computer primary language. It is the only language that the computer understands and does not require interpretation. It is usually written in binary digits (0's and 1's). Machine language is referred to as the first generation programming language because it was the earliest computer programming language.

Low Level Language

a low-level programming language is a programming language that is almost similar to the computer language (machine language). Generally, this refers to either machine code or assembly language. The word "low" refers to the small or nonexistent amount of difference between the language and machine language; because of this, low-level languages are sometimes described as being "close to the hardware". Programs written in low-level languages tend to be relatively non-portable, mainly because of the close relationship between the language and the machine language.

Low-level languages can convert to machine code without a compiler or interpreter. Second-generation programming languages use a simpler translator called an assembler— and the resulting code is understood and executed by the computer machine. A program written in a low level language runs very quickly. An equivalent program in a high-level language can be less efficient and use more memory. Low-level languages are simple, but considered difficult to use, due to numerous technical details that the programmer must remember.

Low-level programming languages are sometimes divided into two categories: first generation and second generation.

High level language

High level language is the programming language written by the programmer in form of English language for better understanding. It uses symbols and words to give instructions to the computer. High level language must also be translated into machine language before execution. It is less machine dependent unlike machine language. The language translator called Compiler, translates high level source statements into machine code at once while the interpreter translates the source program line by line every time the program is executed

Sub-Topic: Types of high level language

There are five types of high level language. They are:

1. Scientific language
2. Multi-purpose language
3. Commercial language
4. Command language for operating system
5. Special purpose programming language.

Scientific Language

This is the language that is used for scientific and engineering purposes. Examples include FORTRAN – Formula Translator, BASIC – Beginner All-purpose Symbolic Instruction Codes. It combines the features of the extensive arithmetic computational ability and the ability to handle mathematical expression.

Multi-Purpose

This type of high level programming language can cope with a number of different types of application areas. Examples of multi-purpose language are Visual BASIC, PL/I C++, Java etc.

Commercial Language

This type of program was first developed by the American Government Defensive Department to create a common administrative language for internal and external use. The prominent language is COBOL (Common Business Oriented Language).

Command for Operating System

This program language is used to control the operating system. An example is DOS Commands (Disk Operating System commands.)

Special Purpose Language

This is programming language designed for a specific assignment. It is tailored towards a particular problem. E.g. Structural Query Language

EXAMPLES OF HIGH LEVEL LANGUAGE

1. BASIC (**B**eginners **A**ll purpose **S**ymbolic **I**nstruction **C**ode)
2. FORTRAN (**FOR**mula **TRAN**slator)
3. PL/1 (**P**rogramming **L**anguage **1**)

4. COBOL (COmmon Business Oriented Language)
5. PASCAL
6. JAVA
7. C – Language etc

Before a program is written, there is need for proper planning on how the problem at hand will be solved. Steps to follow are;

1. Problem definition
2. Planning the solution
3. Program coding
4. Program testing
5. Program documentation

1. **Problem definition:** The programmer is expected to study the problem and also know all inputs to be used in the program and the expected output.
2. **Planning the solution:** Before a program is written, flowchart for that program must first be written and tested before the actual coding of the program.
   A flowchart is a diagrammatical representation of the step by step method involved in writing a given program.
3. **Program coding:** This is the actual coding of the program
4. **Program testing:** The written program is tested and errors are corrected to check the workability of the program
5. **Program documentation:** This involves writing a detailed description about the program, facts pertaining to the usage and maintenance of the program.

EVALUATION

1. What is Computer program?
2. Define programming language
3. Who is a programmer?
4. State differences between Machine language and Low Level Language
5. Mention five (5) types of high level language

READING ASSIGNMENT

Read the next week topic i.e. BASIC Programming.

WEEKEND ASSIGNMENT

List nine (9) BASIC rules

WEEK 9

TOPIC: BASIC PROGRAMMING LANGUAGE

CONTENT:

(a) Basic Language:

(b) Meaning of BASIC

(c) BASIC character set

(d) Key BASIC Statement: i. Line number ii. Remark (REM) iii. Assignment (LET, INPUT, DATA). iv. Output Statement v. Print vi. Program Terminator (END, STOP)

(e) Simple BASIC Statements

Meaning of BASIC

BASIC stands for Beginner All-purpose Symbolic Instruction Code. It is not only simple but also a very powerful high level programming language. **The original BASIC was designed by John Kemeny and Thomas Kurtz in the mid 1960's at Dartmouth College in New Hampshire, USA.** It consists of statements written in English words and mathematical notation. It is written in English-like and mathematical notations.

Rules for BASIC programming

1. All expressions must be written in capital letters.
2. First character must be alphabet
3. BASIC statement or keyword must start with a line number.
4. Each line must contain only one BASIC program statement.
5. There must not be full stop at the end of a statement.
6. The start/begin statement must be the first entry in a program. This does not mean that there is a statement called START.
7. The End statement must be the last entry in a program.
8. Spaces should be inserted to make the program more readable

BASIC Character Set

1. **Character Set:** These are the acceptable characters in BASIC programming. e.g. Numbers 0 - 9, Alphabets A - Z, a – z, Special characters or symbols +, -, *, /, <, &,:,; etc

2. **Data Constants & Data Variables**

i. **Data Constant (or constants)**

These are data that do not change during the course of computation or program execution.

ii. **Data Variables (variables)**

They are data that can change in constant numerals versus variable numerals.

Constant data

```
10      A = 1
20      B = 2
30      Sum = A + B
40      PRINT SUM
50      END
```

Variables data

```
10 INPUT A
20 INPUT B
30 Sum = A + B
40 PRINT SUM
50 END
```

3. **Keyword:**

This is also referred to as a BASIC statement. It is an instruction which has special meaning to the computer or BASIC interpreter. Examples: REM, LET, INPUT, READ, PRINT, GOTO, FOR…., NEXT etc. these must not be used as a variable name during the course of writing any BASIC program.

What is a Variable? A variable is a symbol that can represent any value. Variables are used to represent values needed for processing during program executions. A Variable can be any alphabet or combination of alphabets.

Rules guiding Variable names:

i. It begins with a letter
ii. It does not nave spaces
iii. It should not be the same name as a word already used by the BASIC language such as END, LET, and PRINT etc.

Types of Variables:

i. **Numeric Variable:** these are used to hold numbers, integers, real numbers or exponential form. **Examples are A1, T4, and G9 R2 etc.**

ii. **String Variable ($):**String variables are used to hold inputs that are alphabetic. This kind of variable can be up to 255 characters in length but it must be enclosed in double quotes (“”). It is represented with the **$** sign

Examples of Variable:

Numeric Variable	String Variable ($)
A	A$
GREEN	GREEN$
C1	C1$
DEBORAH	DEBORAH$
ADA	ADA$
L	L$

Sub-Topic 2: Key BASIC Statement

Line Numbers:

In BASIC, we need a line number for each BASIC statement. Line Numbers must be positive whole numbers from 1 to 99,999. A line number is always in integer form and this is done to give room for correction when necessary. It is presented in the format below:

10 ……………………..
20 ……………………..
30 ……………………..
40 …………………….. Etc.

REM Statement (Remark or Remember)

REM statements stand for remark or Remember. The statement allows you to add comment and explanatory notes to your program. This may as well include date and what the program or segment of a program is all about. Computer does not execute REM hence, it is NOT necessary to use Line Number for REM. It is just a remark that aids the programmer or whoever reads the program to remember or understand certain things about the program.

Examples: **REM PROGRAMM TO SAY HELLO**
REM TO CALCULATE AREA OF A TRIANGLE

Sub-Topic 3: Key BASIC Statement (Cont.)

ASSIGNMENT Statement

Assignment statements are used to assign values to a Variable. Examples are **LET, INPUT, and DATA**

INPUT Statement

This allows you to type in data into a variable directly from the keyboard while the program is running. This program will be able to produce the required result with given data. The INPUT statement can be used in the following ways;

Example:

```
10 INPUT "WHAT IS YOUR FIRST NAME?", A$
20 INPUT "WHAT IS YOUR SECOND NAME?", B$
30 INPUT A
```

Note: if you want to a set of written note to display on your output screen exactly the way you have typed it in your program, you enclose the text with a double quote ("").

Example

PRINT "Hello"

LET Statement

The LET statement permits the programmer to assign numbers and formulas to a variable name.

Example:

```
10 LET FISAYO = ½ (b*h)
20 LET J$="How do you do?"
30 LET HEPHZIBAH=50
```

DATA Statement

The statement **(Read and Data)** goes hand in hand. Data statement is used to enter data into a program before running the program or before program execution occurs. The data to be entered into the program is read from DATA statement.

E.g.

```
10 READ A, B, C, D
20 DATA 3, 5, 10, 15
```

Note: the way you arrange the values on DATA line is exactly how the values will go into the variables. So be mindful of it.

OUTPUT Statement (PRINT)

The result of the processed data is displayed by output statement. The PRINT statement brings out the processed data. The print statement has the general format.

Example:

```
20 PRINT D, E, C
25 PRINT A, $, C
30 PRINT “The answer is =”, ADA
```

PROGRAM TERMINATOR (END)

END Statement: this always indicates the end of a BASIC program. When the computer comes across the end statement in a program, the computer automatically ends. End should always be the last statement in the program to indicate the physical end of the program.

END indicates when the logical executions of a program should cease.

Example:

```
    40 END
```

CLS

This means **CL**ear **S**creen. Every BASIC program should begin with CLS to avoid getting unexpected display on the screen.

Sub-Topic 4: Simple BASIC Statements

Structure of BASIC program

- One instruction or statement per line
- Each line must begin with a line number
- Line numbers are unsigned positive integers
- Line number should increase in steps of 10 to allow for insertion of extra lines during program modification.

Example 1: Write a BASIC program to find the **average** of three numbers.

Solution:

```
5       REM FIND AVERAGE
10      READ A, B, C
20      DATA 5, 10, 15
25      LET SUM=A+B+C
30      LET AVE=SUM/3
35      PRINT AVE
40      END
```

Example 2: Write a BASIC program to calculate the volume of a box.
Solution:

```
10      REM FIND THE VOLUME OF A BOX
20      READ L, B, H
30      DATA 3,5,10
40      LET VOLUME =L*B*H
50      PRINT VOLUME
60      END
```

Conditional Statements

These are statements in BASIC programming that are dependent on some certain conditions and criteria before they can be obeyed. Example is IF…THEN…ELSE

IF…THEN…
This is a conditional statement. If the condition is true, the statement following the "THEN" is executed.

Example

```
2 INPUT A
        IF A > 500 THEN GOTO 50 ELSE GOTO 2
50 PRINT A
60 END
```

Sample program

```
10 REM PROGRAM TO PRINT NAME AND ADDRESS
20 LET A$="Deeper Life High School"
30 LET B$="Opete off DSC Express Way, Warri, Delta State"
40 PRINT A$
50 PRINT B$
60 END
```

EVALUATION

1. Write a BASIC program to find the average of five numbers.
2. Write short note on program terminator.
3. Explain the following: Character set, constants and variables

READING ASSIGNMENT

Read the next week topic i.e. Graphic Package – Corel Draw – practice on the use of the tools.

Weekend Assignment

1. _____ always indicates the end of a BASIC program. (a) Start statement (b) Constant (c) END statement (d) Line statement
2. _____ statement is used to enter data into a program before running the program or before program execution occurs. (a) DATA statement (b) Line statement (c) END statement (d) LET statement
3. ____allows you to type in data from the keyboard while the program is running. (a) LET statement (b) DATA statement (c) INPUT statement
(d) Output statement
4. ____allows you to add comment and explanatory notes to your program.
(a) INPUT statement (b) DATA statement (c) REM Statement (d) Output statement
5. ____ consists of statements written in English words and mathematical notation. (a) BASIC (b) FORTRAN (c) Programming (d) Constants

Essay

1. List five rules for BASIC programming.