**COMPUTER SOFTWARE/COMPUTER INTELLIGENCE**

Computer software refers to ***sets electronic instructions***, ***commands*** or ***programs*** which direct ICTs about how to capture data, process, store, and output information (enablers of computer functions).

**Key notes on software;**

* + Software gives the hardware and the computer in general life.
  + ***Software*** defines a ***computers intelligence*** – what a computer can do or be used for.
  + Software are sets of ***programmed instructions.***
  + ***Software i***s produced by a ***programmer/software engineer.*** That is why we say that it is artificial intelligence.
  + There is a software behind every computer activity.

**Characteristics of computer software**

Computer software is described by the following traits;

* + Software **is Intangible**. It is not physical
  + Software **is Engineered**. Its production is a series of logical steps/phases.
    - **Scalability**: software must grow with user or organization’s needs.
  + Software **defines a computers intelligence**. Or what a computer can be used for. It holds the instructions that tell computers what to do.
  + It is **Hackable and can be contaminated** by dirty data and malwares
  + Software does not “**wear out**”
  + Software is User-friendly. Very interactive and can advise the user (intended user) on what to do.
  + Software defines the performance of a computer. It has power

**Software acquisition**

Software is generally got through;

* + - Software writers retail outlets in major cities around the world,
    - Software writer’s agents
    - Software writers certified firms.
    - **Application service providers (ASP):** Third party organizations that manage and distribute software and other computer services online (through the WWW).

**Categories of Software**

Software is generally categorized as;

* + - User-applications.
    - System software/system-applications

**A. User-application software:**

Application software refers to forms of computer software a computer user directly interacts with to do work on and with the computer. It is user dependent.

Forms of User-application Software;

* + - **Customized user-application software.** This is also called **in-house** or **tailor made** or **Bespoke user-applications software.**
    - **Packaged user-application software.** This is also called **off-the-shelves or standardized user-application software**

**Customized User-application Software**

Refers to any software designed for a ***specific ‘person’***

* Customized software can be;
  + - **Vertical customized user-application software** – i.e. if it is used in one department or section of the organization.
    - **Horizontal customized user-application software** – i.e. if it can be used in several or other departments and sections of the company.

**An evaluation of customized user-apps**

**Advantages of customized user-apps**

* It is a bit hard to manipulate.
* It makes the organization using it to look unique. Improve image.
* Customized software applications offer very high productivity for users.
* They usually cater for all organizational needs by providing for all the necessary details.

**Disadvantages of customized user-apps**

* Usually very expensive to design and develop. All production costs are met by one person.
* They are not flexible. Since they can only be used by one person/company.
* They take long time to be acquire.
* They need a lot of specialized skills to use since they are usually not very common.

**Packaged/off-the-shelves /Standardized user-application software**

These are copyrighted and commercial software designed to meet software needs of a wide variety of users.

Packaged software can be;

* **Generalized packaged user-apps –** if they can be used for many tasks like some word processors
* **Specialized packaged user-apps:** if they handle tasks of a specialized nature like accounting, medic, law, architecture and marketing apps.

**Advantages of off-the-shelve user-apps**

* They are always available in software kiosks.
* They are cheap to procure.
* They are very flexible since they can be used by any organization.
* They are the core of entertainment and leisure application programs.
* Do not need a lot of specialized skills.
* They form the majority of educational reference software.

**Disadvantages of off-the-shelve user-apps.**

* They are very easy to manipulate since they are almost known by many people
* They are not very secure.
* May not handle some problem with ease. May not provide the necessary software needs breadth and depth.

**General Common types of application software**

* + Word Processors
  + Spread sheet management apps.
  + Database management apps.
  + Presentation software
  + Integrated software
  + Computer aided design apps.
  + Desktop publishing software
  + Project management apps. – Ms. Project
  + Personal Information Mgt Apps.
  + Video and Audio editing software
  + Multimedia authorizing software
  + Web authoring software
  + Personal finance mgt Apps.
  + Educational Reference software
  + Entertainment software
  + Communications software
  + Accounting software
  + Pointing and Image editing software

**Assignment:** Identify ***four examples*** of each of the above types of software.

***SYSTEM SOFTWARE***

System software relates to all programs designed to help a computer system to manage itself.

Management functions of system software include - ***control***, ***coordination,*** ***planning*** and ***reporting*** of system activities.

System software are to the computer system as user-applications relate to the computer end-user.

System software can also be literally called ***“system-application programs”***

The various ***types of system-applications*** are;

* + - The operating system (**O.S**)
    - Utility or service programs
    - System drivers
    - Firmware
    - Programming tools

**The Operating System**

Is a master control program that manages the ***general*** ***operations*** and ***resources*** of the computer system? For example;

* + - **Android**
    - **Windows family OS like Windows XP, 7, 8, 10, (Windows Millennium) Me, 2000, NT, 95, 3.1, 98, etc.**
    - **Mac**
    - **Linux**
    - **Unix**
    - **Disc Operating system (DOS)**
    - **Wang**

**Classifications of OS**

1. **Multi–tasking OS:** This allows many tasks to be handled by a single CPU system at the a time. ***Multi-tasking*** Operating Systems are ***multi-processing*** and ***multi-user*** Operating Systems
2. **Single user OS:** These are operating systems that allow only one activity in the CPU at a time.

**Functions of the operating system in the computer system**

1. **Manage storage media/allocates memory**: They determine on where data and programs should be placed at the time of processing and after processing.
2. **Management of computer system security:** Enable computer users to create user accounts with passwords.
3. ***Manages system prompts***: They direct or prompt devices and programs to start work.
4. ***Facilitate the booting process:*** Loading the OS into memory is one of the last processes of the booting process. Without a working OS the computer cannot boot. Booting is the process of attaining the operational run-time environment of the computer system. The booting process can be;
   * 1. ***Cold/hard booting:*** Starting the PC when the system has been off.
     2. ***Warm/soft booting:*** Restarting the PC when the OS is already running in memory
5. **Management of system faults and errors:** The OS keeps on checking on system devices and programs. Where it finds an error it reports it to the user or make an effort to fix it. **They monitor system performance and usage**.
6. ***Interface provision:*** The O.S provides tools and mechanisms (interface) through which the user interacts with the computer. Common interfaces provided are;
   * 1. **Menu driven interface – where users interact through lists of options they choose from.**
     2. **Command line interface through command boxes where commands are punched.**
     3. **Voice recognition interface**
     4. **Touch and wave interface**
     5. **Remote terminal or controlled interface**
     6. ***Graphical user interface (*GUI*):*** The OS provides interactive images like –

***dialogue boxes***, ***buttons***, ***icons***, and ***tabs*** with which the user can work with the computer.

**Advantages of GUI**

* + - **Graphical images are easier to learn and work with.**
    - **There is no need to type and memorize any hard command language.**
    - **The interface is similar for any application.**
    - They make program identification easy. Each program has got a unique graphical image

**Disadvantages of GUI**

* + - Graphical images require faster memory and faster processor because they generally heavy.
    - It also occupies more disk space to hold all files for different functions.
    - It is difficult to automate functions for expert users.

1. **Device configuration**. The OS initiates or aligns devices such that they interact well with the entire computer system. Configuration also refers to the alignment or arrangement of the functional part of the computer system.
2. Operating systems provide the platform or foundation into which application programs run.

**SYSTEM UTILITIES**

Utility programs are forms of system software that ***service*** other programs and system devices.

For the body system (***Biology class***), there a number of body utilities like platelets, white and red-blood cells, etc., doing different specialized functions for the body system. Computer system utilities also work like body utilities, but for the computer system.

1. They are also called **Service programs**.
2. ***Some utility*** programs come ***embedded*** in operating systems, while others can be bought independently or down loaded online from the various “***Play-stores***”.

**Classifications and examples of utility programs.**

* **Popular types of utility programs include:**
* **Data recovery –** e.g. Recova. Help in regaining accidently or intentionally deleted data.
* **System restore: They take back the system** to a known previous state that worked well.
* **File compression utilities. They r**educe the size of a file for more storage space to be created on the storage media, & also enable faster file transfer on a network.
* **Diagnostic utility:** compiles technical report about programs and devices . e.g. system restore utility
* **File sorting utilities: They facilitate** rearrangement of files in either descending or ascending order of date, type, name or size.
* **File viewer: They facilitate** copying, pasting and viewing of files.
* ***Backup utility:*** Enables computer users to create extra copies of the same file for reference and security purposes
* ***Disk Scanner:*** They check system disks or storage media for faults/errors. They also make attempts to fixing such the identified errors or faults.
* ***Uninstaller:*** enables users to remove no-longer useful or contaminated programs.
* ***Text editors:*** Enables creation and editing of text based files using basic text formatting features. They do not work with graphics or images.
* ***Recycle bin:*** Contains/holds deleted files and folders
* ***Disc cleanup:*** Enables the user to clear unwanted files and folders.
* ***Antivirus utility:*** They detect, prevent and remove viruses from a computer memory or storage media.
* ***Anti-spyware utilities:*** Prevent system activities from being spied on by third parties.
* **Language translators like compilers and interpreters**
* ***Disk defragmenter:*** reorganizes files and also ensure that bad disc sectors and files are isolated for enhanced performance.
* **Calculators and Calendar utilities**
* ***Screen saver:*** It causes the monitor/screen to display a moving image or blank screen if no keyboard or mouse activity occurs for a specified time period.

**Programming tools**

**Key definitions:**

* ***Programming*** refers to the process of developing computer instructions (programs) to solve a particular task
* **Programming tools** are sets of system programs used to create other programs. They include;
  + - **Programming languages**
    - **Language translators (like interpreters, compilers and assemblers)**
    - **Debugging utilities**
    - **Linkers**
    - **Programming language.** They are platforms or system programs used to create other programs or software

***NB***:

Some programming languages can be acquired independently without other programming tools, while others are sold as a single package including the programming language and associated utilities in one pack called an ***Integrated Development Environment*** (IDE)

**Classifications of programming languages**

* + - ***Low level:*** They are programming platforms where programmer use data binary codes or specific abbreviations when developing programs or software.
      * + **Machine language or 1st generation languages**
        + **Assembly languages or second generation languages**
    - **High level languages: They are programming platforms where programmers use natural words, statements and objects in the program development process. like** C, C++, Pascal, Fortran, BASIC, Java, Lisp, Smalltalk.
    - **BASIC** (Beginning All-purpose Symbolic Instruction Code). This was developed in 1964 by John Kemeny and Thomas Kutz to teach students how to use computers.
    - **FORTRAN**. (FORmula TRANslation). This was developed in 1956 to provide an easier way of writing scientific and engineering applications
    - **(iii) COBOL** (Common Business Oriented Language). Developed for developing business application programs
    - **(iv) PASCAL.** Was developed in early 1970 specifically for computer scientists.
    - Others:
    - **ADA**
    - **ALGOL**
    - **PL/M**
    - (Programming Language Microcomputer)
    - **LOGO**

**OBJECT ORIENTED PROGRAMMING Languages (OOPL)**

* + - **OOPL Use objects to represent data and behavior (like motion). Examples include Visual Basic (C++)**

**Basic Programming Terminologies**

* + - ***Code:*** Is a written program text or statement
    - **Source code:** Is a program code in a language a computer programmer (– the originator or source of the code) understands.
    - ***Object code:*** Is program code which is computer–readable i.e. a source code that has been translated into machine understandable language.
    - ***Translator:*** It is a programing utility that changes or interprets a source code into and object code.

***NB: Computers understand binary and quantum languages where each character (letter, number, shape, light or sound stream) has got its binary code of 8bits or byte, or qubits.***

**Types of Program or language translators**

* + - ***Interpreters*:** utilities that translate program code text file or data statement by statement.
    - ***Compilers:*** translate the entire program code (text file or data) at once.
    - ***Assembler***: An assembler translates a program written in 2nd generation (assembly) language into machine language or first generation language

**NB: In the process of coding and compilation two types of errors are committed as explained below.**

* + - An error in a program or software is technically called a *BUG. (Then, what is a bug in software development and management??)*

**Program codeerrors**

1. **Syntax errors/procedural errors**:

These errors occur as a result of improper use of language rules. Once a syntax error is committed, the written program can't execute or run e.g. language grammar mistakes, punctuation, improper naming of the variables, capitalization in some cases, etc.

**Forms of syntax errors**

* + - **Program grammar errors**
    - **Spelling error *for instance - writing a keyword with wrong spelling***
    - **Punctuation errors or poor punctuation for example - m*issing semicolon to terminate execution of line of code***
    - **Missing Parenthesis e.g. (}) to indicate the start and end of main function.**
    - **Printing the value of variable without declaring it**
    - **Using a function that is not in the included header**
    - **Using wrong case for keywords**
    - **Poor use of space as a character**

1. **Logical errors:**

These are errors which are not detectable by the translator during compilation. The program runs but gives a wrong output.

**Programming error detection methods/approaches**

* + - **Dry run/desk check** – literally meaning physically going through the program code script
    - **Error detection utilities** – like debugging utilities
    - **Use of test data** – giving the program sample data to manipulate

**Basic Programming Terminologies…… Cont’d**

* **Algorithm:** Refers to a limited number of logical steps a program follows to solve a problem. Program algorithm can be represented by;
  + - * **Pseudo code**
      * **Flow chart**
* ***Pseudo code:*** Refers to a set of statements written in a natural readable language (like Lusoga, Luganda, English, French, Lukiga, etc.) but expressing the processing logic of program.

**Sample Pseudo code: A pseudo code for adding and averaging two numbers**

START

Print “Enter two numbers”

Input x,y

Sum – x+y

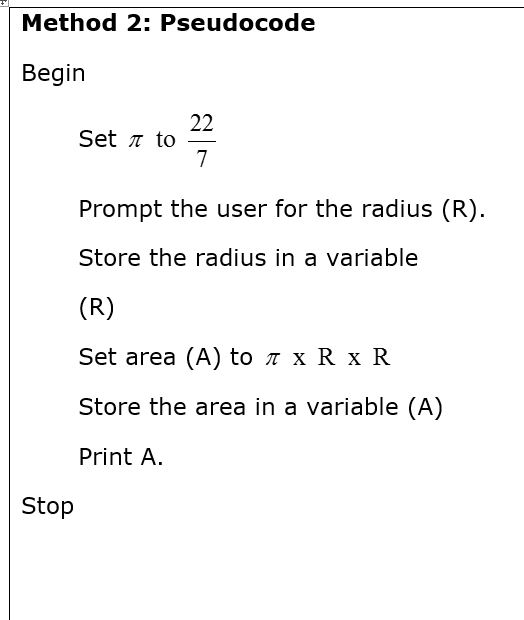
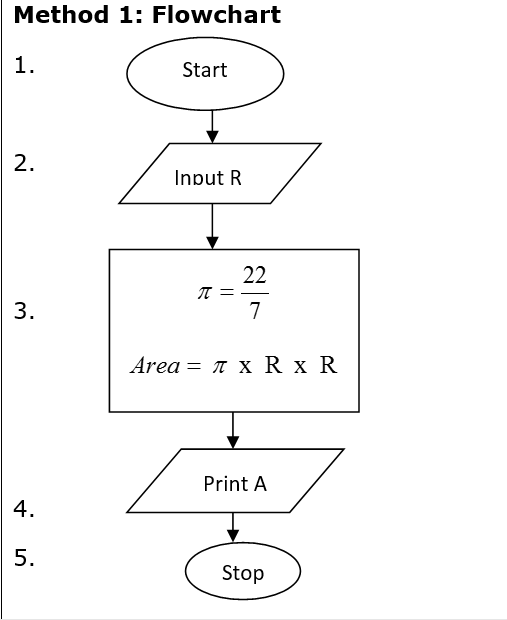
Average = sum /2

PRINT sum

PRINT Average

STOP

**Flow Charts**

A flowchart is a diagrammatic representation of a program’s algorithm

Flow chart construction

* When constructing a flow-chart, we use both statements and special symbols with specific meaning
* The symbols are combined with short text clues which are easily understood by programmers