

COMPUTER SOFTWARE

Computer software refers to the electronic instructions and procedures that command and control the operation of a computer. There are mainly two types of software: **System software** and **Application software**

System Software

System software refers to programs that manage, control and maintain the operation of a computer and all devices connected to it. It performs tasks related to the operation and performance of the computer system and avails resources to the user.

Categories of system software

System software is categorized into four: Operating system, Utility programs, Device drivers and Programming languages.

Operating System software (OS)

Operating system is a set of programs containing instructions that work together to coordinate all the computer activities. It performs daily activities of a computer and manages its resources.

Types of operating system

- Stand-alone operating system
- Network operating system
- Embedded operating system

Common examples of operating systems associated with computers are:

Windows Operating System, Linux, Unix, Ubuntu Operating System, Macintosh (Mac), Chrome Operating System, Novell Operating System, Android For Mobile Phones, etc.

Function of operating system (OS)

- It helps in processor management, in multitasking and time sharing
- Configuring and controlling peripheral devices
- Installing of both system and application software
- Managing system and application programs
- Updating software automatically
- Booting the computer
- Monitoring system performance

- Administering system and network security
- It helps in file management
- Managing memory and storage devices
- Allocating system resources
- Establishing an Internet connection
- Providing the user interface between the user and computer
- Providing a platform for other programs to run
- Responsible for scheduling and coordinating computer jobs
- Spooling print jobs
- Establishing and controlling network connections
- Managing computer errors

Operating System User Interfaces

User interface refers to the visual part of a program or OS through which a user interacts with a computer or software. It determines how commands are given to the computer and how information is displayed on a computer.

There are two forms of operating user interfaces: **Command-line** user interface: and **Graphical** user interface.

Command-line user interface:

In a CLI, a user types commands using a keyboard or every command prompt. The set of commands a user uses to interact with the computer is called **command language**. Examples are DOS, UNIX, and LINUX.

A user is provided with a virtually empty screen with a blinking cursor where commands are keyed and the computer executes them by pressing the enter key.

Advantages of command user interface:

- Commands are executed faster.
- Consumes limited random-access memory
- Cannot easily be affected by viruses • Ideal for programming and programmers •
Commands can be grouped together.

Disadvantages of command user interfaces

- Commands are not easily memorized
- Requires a lot of knowledge to master the commands • Commands have to be typed correctly
- No graphics on the screen.
- A mouse cannot be used

Graphical User Interface (GUI)

In a GUI, a user uses menus and visual images such as icons, buttons and other graphical objects to issue commands. Common GUI operating systems are Windows, Ubuntu, etc.

Advantages of graphical user interface (GUI)

- It is user friendly because it is easy to learn and work with • There is no need to type and memorize any command language.
- The interface is similar for any application.
- You are provided with a coloured screen with icons each representing a program.
- A mouse may be used.

Disadvantages of graphical user interface (GUI)

- Require much more memory and as well as the processor
- Require much more desk space to hold the files
- Difficult to automate for expert users
- Easily affected by viruses
- Not good for programmers.

Utility Software

It is a type of system programs that allows a user to perform maintenance tasks related to nagging a computer, devices connected to it and software installed on it. They are designed to configure, analyze, optimize and maintain a computer in and good operation conditions.

Functions of common utilities

There are a variety of utility programs with specified functions as described below:

1. **File manager** – perform tasks such as displaying file lists on storage media, organizing files in folders, copying, renaming, deleting, moving and sorting files
2. **Search utility** - helps user to locate files on a computer based on a given criterion
3. **Uninstaller** – removes with its associated file from the file system
4. **Image viewer** – allows a user to display, copy and print an image
5. **Disc cleanup utility** – searches for and removes unnecessary files such as downloaded, temporary Internet, deleted files and unused files
6. **Disc defragmenter** – reorganizes files and unused space on a computer hard disc so that data access is quicker, programs run faster
7. **Backup utility** – allows users to copy or backup selected files or an entire hard disc to another storage medium or location

8. **Restore utility** – restores backed up files to their original form in the event of loss of the original copy
9. **Screen saver** – causes the computer screen to display a moving image or blank screen if no keyboard or mouse operation occurs for a specific time. Screensavers are used for security, entertainment, advertisement, etc.
10. **Personal firewall** – detects and protects a computer from unauthorized intrusions. It monitors transmissions to and from a computer
11. **Antivirus utility** – protects a computer by identifying and removing viruses found in memory, storage media, or on incoming files
12. **Spyware remover** – detects and removes spyware and related programs
13. **Adware remover** – a utility that detects and removes adware and related programs
14. **Internet filters** – remove or block certain infected or inappropriate from being received or displayed from the web.
15. **File compressing utility** – shrinks the size of files and folders for easy transfer
16. **Media players** – allows users to view images, animations, play music and videos files on a computer
17. **Disc burning** – writes computer files on optical storage media such as recordable or rewritable CD, DVD or Blue-ray disc.
18. **PC maintenance utility** – identifies and fixes operating system problems, detects and repairs disc problems and improves general system performance.
19. **System archives** - output a stream or a single file when provided with a directory/set of files.
20. **Disk checkers** - scan an operating hard drive for logical (file system) and physical errors
21. **Cryptographic utilities** - used to encrypt and decrypt streams of files.
22. **Sorting utilities** - help to organize data in a given order.
23. **Memory testers** - check for memory failures.
24. **Disk practitioners** - divide a drive into multiple logical drives, each with its own file system which can be mounted by the operating system to be treated as an individual drive.
25. **File synchronization utilities** - maintain consistency between two sources. They enable redundancy or making copies of data.
26. **System profilers** - provide detailed information about the software installed and hardware attached to the computer.
27. **Network utilities** - analyze a computer's network connectivity, configure network settings, check data transfer or log events.
28. **System restore utility**

Device Drivers

A device driver is small software that tells the operating system how to communicate with the device. Each peripheral device on a computer has its own specialized set of commands and thus requires its own specific driver. The operating system loads each device's driver when the computer boots up.

To communicate with the hardware devices, the operating system relies on device drivers. A device driver accepts instructions and then converts them into commands that the device understands.

Programming Languages

These refer to the languages used to write a computer instruction, software. A person who writes a computer program is called a **programmer**.

Functions of software programmer

- He writes a computer program
- He maintains a computer program
- He upgrades a computer

Types of programming languages

Programming languages are of two types

- Low level programming language
- High level programming language

Low level languages

This is a computer language written in machine code language, the language understood by a computer. It is also referred to as binary language because it consists of only two digits, 0s and 1s

Types of low-level languages

- Machine code language (first generation)
- Assembly language (second generation)

Assembly (low-level) languages

This is a language which consist of mnemonic symbols (English like words) used to represent the binary digits of 0s and 1s of machine language. Assembly languages are used to develop system software.

However, computers can only understand machine language, thus assembler languages must be translated into machine language for execution.

Advantages of low-level languages/machine code

- Machine codes are executed faster.
- They don't require either a compiler or interpreters except assembly □ They are suitable to developing operating systems.

Disadvantages of low-level languages

- They are difficult to learn and understand.
- The instructions are expressed in binary form (0 and 1s)
- The programs are machine dependent hence difficult to be used by other machines of different family.

High level programming languages

This is a computer written in official language (known to man) but should be converted to machine code for a computer to understand.

High level languages consist of statements or sequences text including words, numbers and punctuation, much like written natural languages.

Advantages of high-level languages

- They are machine independent and can be used by other computers.
- They are user friendly and problem oriented.
- They are easier to learn, write, correct and revise than assembler and machine languages.
- They can be used for development of application software.

Disadvantages of high-level languages

- They require to be translated to machine code □ They are not executed faster.

Examples of high-level languages

- | | |
|---------------|---------------|
| • COBOL | • Ruby |
| • PASCAL | • C++ |
| • Python | • C#(C-sharp) |
| • Java Script | • PERL |
| • Java | • FORTRAN |
| • C | • PHP |

Language Processors (Translators)

Language processors translate high level languages to low level (machine code) languages which the processor can understand. They are software designed to translate from high level language to low-level language.

Examples of language processors

- A **Compiler** is a program which translates a whole source code from high level language to low level language which can easily be understood by the computer.

- An **Interpreter** translates the source line by line while translating a high-level language to low level language.
- The **Assembler** translates assembler instructions from assembly language to machine code language or the binary code.
- The **linkers** combine compiled programs and determine where they will be located in the memory.
- **Language editors** are applications or software used to write computer language codes. Some also compile. Common language editors, Notepad, Trouble c, Borland, Sublime text, Dev C, C++, Code blocks, Notepad ++ etc.

Common terms used in programming

- **Source code** is an instruction written as text file by the programmer that must be translated by a compiler or interpreter or assembler into an object code before execution.
- Source code cannot be understood by the computer until it has been translated into machine code.
- **Execution** is the process by which a computer system performs the instructions of a computer program.
- **Object code** is a code in machine language that is ready for execution by the computer.
- **Bug** is a logical problem in the source code that stops the software from running give or to give wrong results
- **Debugging** is the process of detecting, checking and identifying problems and errors in the source code.

APPLICATION SOFTWARE

This is software that performs a specific and user function. Application software is installed on operating system. **Application software** consists of programs designed to perform specific tasks for end-users

Types of application software

Application software is categorized into two: *Off-shelf packages* and *Customized* programs, bespoke or tailor-made programs

Off - the - shelf packages

These are programs already written and ready to be run upon their purchase.

Main categories of off shelf packages

1. **Word processors** - programs that allow a user to perform word processing functions, they are document production applications e.g. MS word. A user can use a WP to create, edit, save, format and print letters, reports, and other documents.

2. **Spreadsheet software** – a program with which a user can organize data in rows and columns create graphs and can perform calculations e.g. MS excel.
3. **Presentation software** - an application with which a user can organize content on a slide for viewing by the audience e.g. MS Power point
4. **Database software** – an application with which one can store lots of data for future manipulation
5. **Desktop publishing software** – software, with which one can design publications like newspapers, wedding cards certificates etc.
6. **Imaging software** - application software with which users can edit, retouch a photograph and images
7. **Web browsing software** - software used for displaying and viewing webpages from the Internet or html documents on computers. Examples include, Mozilla firebox, Internet explorer, Safari, Opera, Netscape navigator, Chrome
8. **Web authoring software** – these are used by webmasters for building websites. Examples include: Microsoft FrontPage, Adobe Dreamweaver, Microsoft expression web, Antenna web design studio, Sublime text 3
9. **Media players for audio and video** – used for audio and video playback on computers. Examples include: Windows media player,, Nero ShowTime, Jet-audio, Power dvd, VLC media player, Virtual DJ, etc.
10. **Graphics software** - used by graphic designers to create and design artistic graphics and to manipulate visual images on a computer such as Logos, cartoons etc. Examples include Paint, Adobe photo shop, Corel draw, Adobe illustrator etc.
11. **Accounting software** - helps companies to record and report their financial transactions. With accounting soft-ware, you perform accounting activities related to the general ledger, accounts receivable, accounts payable, purchasing, invoicing, job costing, payroll functions, etc. Examples include quick books
12. **Audio and video editing software** - lets users produce studio quality soundtracks. Video editing software is used to modify video clips: you can reduce the length of a video clip, reorder a series of clips, or add special effects such as words that move horizontally across the screen etc. common audio and video editing software are:
 - Adobe premiere video editor
 - Ulead video editor
 - Adobe audition for –audio

- Fruity loops studio-audio

13. Images viewing software - computer applications primarily used for previewing digital photographs on the computer screen. Many have basic features such as viewing thumbnails, slideshows, printing and simple editing such as cropping and resizing.

Examples of image viewers include: Microsoft office picture manager, Windows photo viewer, Picasa photo viewer, etc.

14. Reference software - provides valuable and thorough information for all individuals. Popular reference software includes encyclopedias, dictionaries, health/medical guides, and travel directories. Examples include: Encyclopedia Britannica 2011 Ultimate Reference DVD, Microsoft Student with Encarta Premium, etc.

15. Note taking software - enables users to enter typed text, handwritten comments, drawings, or sketches anywhere on a page and then save the page as part of a notebook. the software can convert handwritten comments to typed text or store the notes in handwritten form. Examples include Microsoft Office OneNote

16. Text editors - simple word processors that are generally used to type without any special formatting, mainly used to create small notes, memos and programs. Examples of common text editors are: Notepad, Notepad++, Sublime Text, Gedit etc.

17. Gaming software - programs developed as electronic game that involve human interaction with a user interface to generate visual feedback on a computer. Common computer games include Solitaire, Chess Titans, Racing, StarCraft, Need for Speed, GTA vice City, and Alien Shooter.

18. Email software - is a computer program used to access and manage a user's email account. Web applications that provide message management, composition, and reception functions are sometimes also commonly referred to as webmail. Common ones are Microsoft outlook, Pegasus mail, Mozilla's thunderbird, Kmail, evolution and Apple mail. Popular web-based email clients include: Gmail, yahoo Mail, mail.com, Lycos mail, and Hotmail.

Advantages of off-shelf packages

- They are relatively cheaper than bespoke packages.
- They are readily available for purchase.
- They are user friendly; many have a graphical user interface, sample manuals provided on purchase.
- The user does not face research and development costs or problems since the package is already in operation.
- The user faces limited risk since the user has an option to seek information from existing users about the package before making a decision to purchase.

Disadvantages of off-shelf packages

- They hardly full fill users' specific needs perfectly.
- They are standardized implying it may be hard to make adjustments.
- They may not be recommended where a particular company wants to gain competitive advantage over others.
- Where the package does too much compared to the company needs, the amount of extra space occupied in the hardware can lead to memory wastage.
- The package may also be limited to a particular operating system
- It may be easy to manufacture viruses that may tamper with such application programs.

Customized (Tailor-Made) Or Bespoke programs

These are tailor made programs constructed to meet specific user requirements. They are specifically designed to meet user requirements.

Merits of customized programs

- Ability to satisfy user's specific needs
- The company will be able to perform tasks with its software that its competitors cannot do with theirs thereby gaining a competitive advantage.
- They can easily be modified or upgraded.
- Ownership is to the company that ordered for the software.

Disadvantages of customized software

- Time taken to develop the package may be long yet requirements could be urgent.
- The cost of paying programmers makes them expensive. The organisation has to incur development and consultancy costs in this case.
- They are not flexible i.e. not meant to do various tasks.
- Expensive to maintain. When they breakdown programmers who are expensive have to be called in
- In-case of breakdown, the company may be brought to a standstill □ There is a greater probability of bugs in bespoke programs.
- They are not compatible with all computer types, designed for specific computer types.
- They may not run which means that the programmer has to design another one. This could even lead to more costs on the side of the user as well as causing delays

Shareware

Shareware is copyrighted software that is distributed at no cost for a trial period. To use a shareware beyond that period, you send payment to the developer. In some cases, a scaled-down version of the software is distributed free, and payment entitles the user to the fully functional product.

Copyrighted software

Copyrighted software refers to computer programs with restrictions regarding use, modification, and redistribution. You have to pay for copyrighted software and must not copy it without permission from the manufacturer. Copying copyrighted software without paying for it is clearly unethical and illegal.

Freeware

Freeware is copyrighted software provided at no cost by an individual or a company that retains all rights to the software. Therefore, other programmers cannot include freeware in applications they intend to sell.

Open-source

Open-source software is software provided for use, modification, and redistribution. This software has no restrictions from the copyright holder. Open-source software usually can be downloaded from the web at no cost.

Public-domain software

Public-domain software has been donated for public use and has no copyright restrictions. Anyone can copy or distribute public domain software to others at no cost.

Web-based software

Web-based software refers to programs hosted by a web site. Users access and interact with webbased software from any computer or device that is connected to the Internet. Many web sites allow free access to their programs; some charge a fee. Examples of web-based software include e-mail, Website builders, online games, travel and mapping software, etc.

Special purpose (specialized) software

This refers to computer programs developed and dedicated to particular jobs only. Programs that run on special purpose computers like ATMs are special purpose software.

Other examples of specialized software include:

- Business – transaction and sales management software.
- Science and engineering software etc.

General purpose

This refers to a wide a variety of application programs that perform many common tasks. Varieties of general-purpose application programs include word processing programs, spreadsheet programs, web browsers, graphics programs, etc.

Software suite

A software suite is a collection of several applications that are bundled together and sold or distributed as a single package. The software programs may have correlative features and functionality or they may be completely different from one another but share a similar theme. When you install the suite, you install the entire collection of applications at once instead of installing each application individually. At a minimum, suites typically include the following software applications: word processing, spreadsheet, database, and presentation graphics,

Examples of software suites

- Microsoft office • Quick office & office
- Libre office productivity suite. suite
- Open office.org • Prooffice free 3.0
- Word perfect office x5 • K-office
- Zoho • Lotus smart suite
- Adobe master suite

Advantages of using software suites

- Costs significantly cheaper than buying each of the application package separately
- Easy to learn and use because applications within a suite usually use a similar interface and share common features such as clip art and toolbars.
- Easy installation because all the various applications can be installed at once.

Common terminologies used in software

1. **Software license** - This is a document that provides legally binding guidelines for the use and distribution of software.
2. **Software agreement** - This refers to the legal contact between licensor and /or author and the purchaser of a piece of software which establishes the purchaser's' rights.
3. **Software piracy** this is the illegal duplication of copyrighted software.
4. **Software bug** refers to an error in the programming code that does not permit it to function well.
5. **Beta software** is a type of software provided to people for testing purposes
6. **Software release** is the process of issuing/letting the software or application for publication, use and distribution.

7. **Software version** refers to variation of an earlier or original type with minor changes to the existing version or type. E.g. iTunes 12.0, iTunes 12.2.3
8. **Hot fix**: this is software that is designed to fix a bug or security hole in software program.
9. **Spy ware**: this is a software application that is designed to gather information about a person or organization without their knowledge that may send such information to another entity.
10. **Software patch**: this is software that is designed to modify, correct, and fix problems in software.
11. **Software update**: this is a software application that provides fixes for features that are not working as intended or adds minor software enhancements and compatibility.
12. **Software upgrade**: this is a process of replacing a product with a newer version of the same product.
13. **Firmware**: this is a permanent software that is programmed or embedded in a hardware usually read-only-memory by the manufacturer.

Qualities of a Good Software package

- Should be error free-operate without bugs
- Should provide help in case of user errors
- Should be compatible or machine independent
- Should not stop working no matter the errors made by the user-robust
- Should be easy to use, learn with simple user interface
- Should be easy to update, upgrade and maintain
- Should have user manuals and documentation
- Should be reliable, predictable, reliable and dependable.
- Should be capable of displaying error messages to the users
- Should be scalable-easy to change to meet user needs
- Should be interoperable-work with other software
- Should be accurate in retrieving results
- Should be portable-work on multiple operating systems
- Should be efficient to meet the users' targets
- Should be readily available
- Should be authentic, genuine, licensed
- It provides the required functionality.
- It has a life-time (measured in years).
- It is accompanied by complete documentation.
- It can be easily customized, configured.

Factors to consider before obtaining a software program

- **Correctness** — does the software do what it is supposed to do (according to the design specs)?
- **Robustness** — how does the software respond to unexpected conditions (wrong input)?
- **User-friendliness** — is the software easy to use by users from the intended audience?
- **Adaptability** — how difficult is it to modify the software to adjust to an ever-changing world?
- **Cost effectiveness**

Characteristics of good software

- **Suitability.** This is the essential functionality characteristic and refers to the appropriateness (to specification) of the functions of the software.
- **Accurateness.** This refers to the correctness of the functions, an atm may provide a cash dispensing function but is the amount correct?
- **Interoperability.** A given software component or system does not typically function in isolation. It concerns the ability of a software component to interact with other components or systems.
- **Compliance.** Where appropriate certain industry (or government) laws and guidelines need to be complied with. This s addresses the compliant capability of software.
- **Security.** This relates to unauthorized access to the software functions.
- **Fault tolerance.** The ability of software to withstand (and recover) from component, or environmental, failure.
- **Understandability.** Determines the ease of which the systems functions can be understood, relates to user mental models in human computer interaction methods.
- **Learnability.** Learning effort for different users, i.e. Novice, expert, casual etc.
- **Maintainability.** Characterizes the amount of effort to change a system.
- **Stability.** Characterizes the sensitivity to change of a given system that is the negative impact that may be caused by system changes.
- **Replace ability.** Characterizes the plug and play aspect of software components, that is how easy is it to exchange a given software component within a specified environment.