**SOFTWARE ENGINEERING**

**26nd September 2020**

# Computer Introduction:

# What is a computer?

Computer is an advanced electronic device that takes raw data as input from the user and processes these data under the control of set of instructions (called program) and gives the output and saves output for the future use. It can process both numerical and non-numerical calculations.

Computer has input device and output devices they are :

* Input devices : user give instruction to the computer
* Keyboard
* Mouse
* Scanner
* Touch screen
* Output devices : user get the result.
* Monitor
* Printer
* Speakers

## **Classified into two types :**

### **Computer hardware**

Computer hardware is physical components that computer system requires to function , it encompasses everything with a circuit boars that operates within a pc or laptop .

### **Computer software**

Computer software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself.

Computer inner parts are cpu , motherboard , hard drive , ram , rom etc.

Booting process like when we power on the computer then its start booting procees operating system comes from hard drive to ram then its on the computer (show the software ).

# What are the different components of a computer? Explain in detail with examples.

computers follow the same basic logical structure and perform the following five basic operations for converting raw input data into information useful to their users.

# Input unit

This unit contains devices with the help of which we enter data into the computer. This unit creates a link between the user and the computer. The input devices translate the information into a form understandable by the computer.

# CPU

CPU is considered as the brain of the computer. CPU performs all types of data processing operations. It stores data, intermediate results, and instructions (program). It controls the operation of all parts of the computer.

## **Memory unit**

This unit can store instructions, data, and intermediate results. This unit supplies information to other units of the computer when needed. It is also known as internal storage unit or the main memory or the primary storage or Random Access Memory (RAM).Its size affects speed, power, and capability

## **Control Unit**

This unit controls the operations of all parts of the computer but does not carry out any actual data processing operations.

* It is responsible for controlling the transfer of data and instructions among other units of a computer.
* It manages and coordinates all the units of the computer.

## **ALU (Arithmetic Logic Unit)**

This unit consists of two subsections namely,

* Arithmetic Section
* Logic Section

### **Arithmetic Section**

Function of arithmetic section is to perform arithmetic operations like addition, subtraction, multiplication, and division. All complex operations are done by making repetitive use of the above operations.

### **Logic Section**

Function of logic section is to perform logic operations such as comparing, selecting, matching, and merging of data.

# Output Unit

The output unit consists of devices with the help of which we get the information from the computer. This unit is a link between the computer and the users. Output device translate the computer's output into a form understandable by the users.

# What is computer operating system? Explain different types of them examples.

An **Operating system (OS)** is a software which acts as an interface between the end user and computer hardware. Every computer must have at least one OS to run other programs. An application like Chrome, MS Word, Games, etc needs some environment in which it will run and perform its task. The OS helps you to communicate with the computer without knowing how to speak the computer's language. It is **not** possible for the user to use any computer or mobile device without having an operating system.

Here is a list of Operating Systems **window , linux , ios , android.**

|  |
| --- |
| USER APPLICATION OPERATING SYSTEM HARDWARE |

## Features of Operating System

* Protected and supervisor mode
* Program Execution
* Memory management Virtual Memory Multitasking
* Handling I/O operations
* Error Detection and handling

## Types of Operating system

* Batch Operating System
* Multitasking/Time Sharing OS
* Multiprocessing OS
* Real Time OS
* Distributed OS
* Network OS
* Mobile OS

# Explain computer software and hardware with examples.

## computer software

Software is a set of programs, which is designed to perform a well-defined function. A program is a sequence of instructions written to solve a particular problem.

There are two types of software − System Software , Application Software.

### System Software

The system software is a collection of programs designed to operate, control, and extend the processing capabilities of the computer itself. System software is generally prepared by the computer manufacturers. These software products comprise of programs written in low-level languages. System software serves as the interface between the hardware and the end users.

Some examples of system software are Operating System, Compilers, Interpreter, Assemblers, etc.

### Application Software

All software applications prepared in the computer lab can come under the category of Application software.Application software may consist of a single program, such as Microsoft's notepad for writing and editing a simple text. It may also consist of a collection of programs, often called a software package, which work together to accomplish a task, such as a spreadsheet package.

some examples of application software are Microsoft WordMicrosoft , ExcelMicrosoft , PowerPoint , Income Tax Software , Railways Reservation Software.

## computer hardware

Hardware represents the physical and tangible components of a computer, i.e. the components that can be seen and touched.

Examples of Hardware are the following −

**Input devices** − keyboard, mouse, etc.

**Output devices** − printer, monitor, etc.

**Secondary storage devices** − Hard disk, CD, DVD, etc.

**Internal components** − CPU, motherboard, RAM, etc.

# Programming Languages Introduction:

# What is a programming language?

Computer programming languages allow us to give instructions to a computer in a language the computer understands. Just as many human-based languages exist, there are an array of computer programming languages that programmers can use to communicate with a computer. The portion of the language that a computer can understand is called a “binary.” Translating programming language into binary is known as “compiling.” Each language, from C Language to Python, has its own distinct features, though many times there are commonalities between programming languages.

These languages allow computers to quickly and efficiently process large and complex swaths of information.

# Explain different types of programming languages with examples.

## python

Python is an advanced programming language that is interpreted, object-oriented and built on flexible. Web and Internet development (frameworks, micro-frameworks and advanced content management systems); scientific and numeric computing; desktop graphical user interfaces (GUIs)

Simple to learn and easily read.Associated web frameworks for developing web-based applications

## JAVA

Java is a general-purpose, object-oriented, high-level programming language with several features that make it ideal for web-based development.

Java is used to develop enterprise-level applications for video games and mobile apps, as well as to create web-based applications with JSP (Java Server Pages). When used online, Java allows applets to be downloaded and used through a browser, which can then perform a function not normally available.

## HTML

HTML is the standard markup language used to create web pages; it ensures proper formatting of text and images (using tags) so that Internet browsers can display them in the ways they were intended to look. Easy to use and learn the basics of HTML

## C

C Language is a structure-oriented, middle-level programming language mostly used to develop low-level applications.

C Language is used to develop systems applications that are integrated into operating systems such as Windows, UNIX and Linux, as well as embedded softwares. Applications include graphics packages, word processors, spreadsheets, operating system development, database systems, compilers and assemblers, network drivers and interpreters.

## PHP

PHP is an open-source scripting language designed for creating dynamic web pages that effectively work with databases. It is also used as a general-purpose programming language.

Easy to learn (as simple as embedding code inside HTML)

Free and opensource

Can be used on all major operating systems and web servers

# What is a compiler and an interpreter. Explain in detail with examples.

**compiler:**  
It is a translator which takes input i.e., High Level Language and produces an output of low level language i.e. machine or assembly language.

* Compiler is more intelligent than an assembler it checks all kinds of limits, ranges, errors etc.
* But it’s program run time is more and occupies a larger part of memory. It has slow speed because a compiler goes through the entire program and then translates the entire program into machine codes.
* As it scans the code in one go, the errors (if any) are shown at the end together.
* Programming languages like C, C++, Java use compilers.

## **interpreter:**

An interpreter is a program which translates a programming language into a comprehensible language. –

* It translates only one statement of the program at a time.
* Interpreters, more often than not are smaller than compilers
* Considering it scans code one line at a time, errors are shown line by line.
* Programming languages like JavaScript, Python, Ruby use interpreters.

# Software Engineering Introduction:

# What is software engineering? What are the different phases involved in software engineering?

Software engineering is a detailed study of engineering to the design, development and maintenance of software. Software engineering was introduced to address the issues of low-quality software projects. Problems arise when a software generally exceeds timelines, budgets, and reduced levels of quality. It ensures that the application is built consistently, correctly, on time and on budget and within requirements.

A software product is judged by how easily it can be used by the end-user and the features it offers to the user. An application must score in the following areas:-  
  
1) Operational: -This tells how good a software works on operations like budget , usability, efficiency, correctness ,functionality , dependability , security and safety.  
  
2) Transitional: - Transitional is important when an application is shifted from one platform to another. So, portability, reusability and adaptability come in this area.  
  
3) Maintenance: - This specifies how good a software works in the changing environment. Modularity, maintainability, flexibility and scalability come in maintenance part.

# What is a software development life cycle? Explain different types of software development life cycles with examples?

SDLC or the Software Development Life Cycle is a process that produces software with the highest quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases that help an organization to quickly produce high-quality software which is well-tested and ready for production use.

## Waterfall Model

This SDLC model is the oldest and most straightforward. With this methodology, we finish one phase and then start the next. Each phase has its own mini-plan and each phase “waterfalls” into the next. The biggest drawback of this model is that small details left incomplete can hold up the entire process.

## Agile Model

The Agile SDLC model separates the product into cycles and delivers a working product very quickly. This methodology produces a succession of releases. Testing of each release feeds back info that’s incorporated into the next version. According to Robert Half, the drawback of this model is that the heavy emphasis on customer interaction can lead the project in the wrong direction in some cases.

## Iterative Model

This SDLC model emphasizes repetition. Developers create a version very quickly and for relatively little cost, then test and improve it through rapid and successive versions. One big disadvantage here is that it can eat up resources fast if left unchecked.

## V-Shaped Model

An extension of the waterfall model, this SDLC methodology tests at each stage of development. As with waterfall, this process can run into roadblocks.

## Big Bang Model

This high-risk SDLC model throws most of its resources at development and works best for small projects. It lacks the thorough requirements definition stage of the other methods.

## Spiral Model

The most flexible of the SDLC models, the spiral model is similar to the iterative model in its emphasis on repetition. The spiral model goes through the planning, design, build and test phases over and over, with gradual improvements at each pass

# Compare different types of software development life cycles.

## Advantages of the waterfall model

1. Simple to understand and use.
2. Each phase is independent of other phases and is processed and completed separately.
3. Suitable for smaller projects, and for projects where the requirements are clearly outlined.

Disadvantages of the waterfall model

1. No output or working software is produced until late in the life cycle.
2. High degree of uncertainty and risks.
3. Not a good choice for big or ongoing projects.

## Advantages of the iterative model

1. Produces working software early during the lifecycle.
2. More flexible as scope and requirement changes can be implemented at low cost.
3. Testing and debugging is easier, as the iterations are small.
4. Low risks factors as the risks can be identified and resolved during each iteration.

Disadvantages of the iterative model

1. This model has phases that are very rigid and do not overlap.
2. Not all the requirements are gathered before starting the development; this could lead to problems related to system architecture at later iterations.

## Advantages of the spiral model

1. Good for large and critical projects
2. Working software is produced early during the lifecycle.
3. Large amount of risk analysis.

Disadvantages of the spiral model

1. Involves higher cost.
2. Not suitable for smaller projects.
3. Project success depends on the risk analysis phase - hence, it requires highly specific expertise in risk analysis.

## Advantages of the prototype model

1. Benefits from user input.
2. As a working model of the system is provided, users get a better understanding of the system that is being developed.
3. Errors and risks can be detected at a much earlier stage, as the system is developed using prototypes.

Disadvantages of the prototype

1. Increases complexity of the overall system.
2. Involves exploratory methodology and therefore involves higher risk.
3. Involves implementing and then repairing the way a system is built, so errors are an inherent part of the development process.

# Python Introduction:

# What is PYTHON programming language? Give a summary about its origin and applications.

**Python** is an interpreted, object-oriented, high-level programming language with dynamic semantica. It is used in web development, data science and so on. for beginners, Python has simple easy-to-use syntax. This makes Python an excellent language to learn to program for beginners.

Python language is created by  **Guido Van Rossum**  in the late 1980s. It is a very simple but powerful high-level programming language making it the best language for beginners to learn programming. The creation of the language is inspired by the programming language ABC which had been developed in the Netherlands at the CWI (Centrum Wiskunde & Informatica).

**Applications**

Python is known for its general-purpose nature that makes it applicable in almost every domain of software development. Python makes its presence in every emerging field. It is the fastest-growing programming language and can develop any application

* Web Applications
* Desktop GUI Applications
* Software Development
* Image Processing Application