

MODULE - 1

[1] What is Software?

"Software is a set of programs (sequence of instructions) that allows the users to perform a well-defined function or some specified task."

Software is a set of instructions, data or programs used to operate computers and execute specific tasks. It is the opposite of hardware, which describes the physical aspects of a computer. Software is a generic term used to refer to applications, scripts and programs that run on a device. It can be thought of as the variable part of a computer, while hardware is the invariable part.

Software is responsible for directing all computer-related devices and instructing them regarding what and how the task is to be performed. However, the software is made up of binary language (composed of ones and zeros), and for a programmer writing the binary code would be a slow and tedious task. Therefore, software programmers write the software program in various human-readable languages such as Java, Python, C#, etc. and later use the source code.

Types of Software

Software's are broadly classified into two types, i.e., System Software and Application Software.

1) System Software

- System software is a computer program that helps the user to run computer hardware or software and manages the interaction between them. Essentially, it is software that constantly runs in the computer background, maintaining the computer hardware and computer's basic functionalities, including the operating system, utility software, and interface. In simple terms, you can say that the system acts as a middle man that checks and facilitates the operations flowing between the user and the computer hardware.
- System software is not limited to the operating system. They also include the basic I/O system procedures, the boot program, assembler, computer device driver, etc. This software supports a high-speed platform to provide effective software for the other applications to work in effortlessly. Therefore, system software is an essential part of your computer system. They are the first thing that gets loaded in the system's memory wherever you turn on your computer. System software is also known as "low-level software" because the end-users do not operate them.

- System software is software that directly operates the computer hardware and provides the basic functionality to the users as well as to the other software to operate smoothly. Or in other words, system software basically controls a computer's internal functioning and also controls hardware devices such as monitors, printers, and storage devices, etc. It is like an interface between hardware and user applications, it helps them to communicate with each other because hardware understands machine language (i.e. 1 or 0) whereas user applications work in human-readable languages like English, Hindi, German, etc. so system software converts the human-readable language into machine language and vice versa.

Features of system software:

- System Software is closer to the computer system.
- System Software is written in a low-level language in general.
- System software is difficult to design and understand.
- System software is fast in speed (working speed).
- System software is less interactive for the users in comparison to application software.

Types of system software:

1. Operating System:

It is the main program of a computer system. When the computer system ON it is the first software that loads into the computer's memory. Basically, it manages all the resources such as memory, CPU, printer, hard disk, etc., and provides an interface to the user, which helps the user to interact with the computer system. It also provides various services to other computer software. Examples of operating systems are Linux, Apple macOS, Microsoft Windows, etc.

2. Language Processor:

As we know that system software converts the human-readable language into a machine language and vice versa. So, the conversion is done by the language processor. It converts programs written in high-level programming languages like Java, C, C++, Python, etc (known as source code), into sets of instructions that are easily readable by machines (known as object code or machine code).

3. Device Driver:

A device driver is a program or software that controls a device and helps that device to perform its functions. Every device like a printer, mouse, modem, etc. needs a driver to connect with the computer system externally. So, when you connect a new device with your computer system, first you need to install the driver of that device so that your operating system knows how to control or manage that device.

2) Application Software

- Application programs or software applications are end-user computer programs developed primarily to provide specific functionality to the user. The applications programs assist the user in accomplishing numerous tasks such as doing online research, completing notes, designing graphics, managing the finances, watching a movie, writing documents, playing games, and many more. Therefore, many software applications are designed and developed every year by companies as per the demand and requirements of the potential users. The application software can either be designed for a general-purpose or specially coded as per the requirements of business cooperation.
- Software that performs special functions or provides functions that are much more than the basic operation of the computer is known as application software. Or in other words, application software is designed to perform a specific task for end-users. It is a product or a program that is designed only to fulfill end-users requirements. It includes word processors, spreadsheets, database management, inventory, payroll programs, etc.

Features of application software:

- An important feature of application software is it performs more specialized tasks like word processing, spreadsheets, email, etc.
- Mostly, the size of the software is big, so it requires more storage space.
- Application software is more interactive for the users, so it is easy to use and design.
- The application software is easy to design and understand.
- Application software is written in a high-level language in general.

Types of application software:

1. General Purpose Software:
This type of application software is used for a variety of tasks and it is not limited to performing a specific task only. For example, MS-Word, MS-Excel, PowerPoint, etc.
2. Customized Software:
This type of application software is used or designed to perform specific tasks or functions or designed for specific organizations. For example, railway reservation system, airline reservation system, invoice management system, etc.
3. Utility Software:
This type of application software is used to support the computer infrastructure. It is designed to analyze, configure, optimize and maintains the system, and take care of its requirements as well. For example, antivirus, disk fragmenter, memory tester, disk repair, disk cleaners, registry cleaners, disk space analyzer, etc.

Difference between system software and application software

System Software	Application Software
It is designed to manage the resources of the computer system, like memory and process management, etc.	It is designed to fulfill the requirements of the user for performing specific tasks.
Written in a low-level language	Written in a high-level language
Less interactive for the users	More interactive for the users
System software plays vital role for the effective functioning of a system.	Application software is not so important for the functioning of the system, as it is task specific.
It is independent of the application software to run.	It needs system software to run.

[2] What are the types of Applications?

The application software consists of three categories, which are: General Application Software, Customized Application Software, Business Application Software

1) General Application Software

1. Word Processing Software

Word processing software is used to format, beautify, and manipulate text. It allows features such as synonyms and antonyms. You can change the fonts, colors, and style according to your choice with the word art feature. Error checking as well as grammar and spell-checking options are also available in it. Microsoft Word is the best example of a word processing software.

2. Spreadsheet Software

Spreadsheet software is majorly used to store data in table format and perform calculations. Intersecting cells are given in a spreadsheet to keep various data fields such as time, date, text, and numbers. Users can perform calculations with formulas and functions. The best example of spreadsheet software is Microsoft Excel.

3. Presentation Software

Presentation Software lets you put forth your thoughts and ideas in a piece of visual information. Then, you can present that information in the form of slides. You can make your slides interactive and informative by adding videos, texts, charts, graphs, and images. The best example of presentation software is Microsoft PowerPoint.

4. Multimedia Software

Multimedia Software lets you create or record videos, audio, and image files. Such app software is used in video editing, graphics, and animations. Common examples of multimedia software are VLC player, MX Player, and Windows Media Player.

5. Web Browsers

These software applications are used to browse on the internet. They let you locate and retrieve data from the web. The most popular web browsers are Chrome and Firefox.

6. Educational Software

These types of application software are called academic software as they are particularly designed to facilitate learning. All different kinds of tutorial software are included in it. Examples of educational software are EDX, MindPlay, and Kid Pix.

7. Graphics Software

Graphics Software is used to make changes in visual data, images, and animation. It comprises different editorial software. Adobe Photoshop, Unity 3d, and PaintShop are examples of graphics software.

8. Freeware

As you can guess from the name, this type of software is available free of cost. Therefore, you can download and install them for free. However, you are not allowed to make any change in its source code. Skype is an example of freeware software.

9. Shareware

Such softwares are distributed to the users on a trial basis. Then, if the users like it and want to continue, they have to pay for that software. An example of shareware software is Winzip.

10. Simulation Software

Simulation Software is a monitoring program that allows the user to observe an operation without performing it. Such software is useful when the existing system's work is not highly accurate, predictable, or dangerous. It is used widely in engineering, robotics, flight systems, weather forecast, testing, education, and video gaming. MATLAB is the best example of simulation software.

11. Open Source

Open Source software is available with a source code and rights for anyone to inspect, modify and enhance it. Moreover, most open-source software is available for free, while very few are paid ones at such a conditional level.

12. Closed Source

Closed Source software is precisely the opposite to open-source software. They are paid software and have intellectual property rights or patent over source code. It usually comes with some restrictions as well as terms and conditions.

2) Customized Application Software

- To remain alive in this competitive market, nowadays, many organizations go for application software development to build application software for business based on their specific requirements. However, these types of software applications are developed on their special demands. This application program also excludes the extra functionalities of application software. Therefore, these functions of application software are available, which customers actually want for their purpose. However, we can't find this application software for business anywhere on the internet. So, we can't download directly from anywhere.
- Furthermore, if you hire software developers to develop a computer application, they will be responsible for handing it over to you with certain rights and terms. Thus, it's also referred to as customized application software or tailor-made software. For example, the computer application or application software used for the salon comes under this category.

3) Business Application Software

- Business application software is one type of application software that aims to fulfill multiple functions and operations of businesses. Utilizing these applications, businesses can improve the accuracy and efficiency of operations. Moreover, the software application ensures the security of systems and programs throughout the operations. Businesses nowadays use many business applications. The following are some of them:
 - Project Management Software Application like Slack, Asana, Basecamp
 - Communications Application Software like Buffer, HootSuite, Zoom
 - Enterprise Resource Planning (ERP) Application Software like Odoo, Oracle, Microsoft Dynamics
 - Customer Relationship Management (CRM) Application Software like Mailchimp, Zoho, NetSuite
 - Database like MySQL, Microsoft SQL, MongoDB
 - Education Software like Skill Lake, Google Classroom, Litmos
 - Human Resource Management Software (HRMS) like BambooHR, Gusto HR

[3] What is Programming?

- In computer science fields, the word program characterizes what a computer actually does and this process is known as programming. We can also define the term programming as **it is the process that models or structure the set of instructions that instruct the machine how to perform a task and what to perform**. It can be done using a variety of programming languages such as C, C++, C#, Python, Java, JavaScript, Ruby, PHP etc.

A computer program consists of code that is executed on a computer to perform particular tasks. This code is written by programmers.

Programming is the process of giving machines a set of instructions that describe how a program should be carried out. Programmers will spend their whole careers learning a variety of programming languages and tools so they can effectively build computer programs.

- Programmers will start by using a code editor or IDE to write what is called source code. This is a collection of code written in a programming language that other programmers can read.
- Source code needs to be converted into machine language so machines can understand the instructions and execute the program. This process of converting source code into machine language is known as compiling. Examples of compiled programming languages would be C and C++.
- There are other languages that do not use compilers. Instead, these languages will use an interpreter that will read and execute the code. Examples of interpreted programming languages would be JavaScript and PHP.
- Once the code is executed, then the computer program can run. The different types of computer programs include Word processors, Database systems, video games, and websites.
- These computer programs allow us to interactive with different software devices and services like phones, websites, and the computers themselves.
- Some of these languages are primarily used in one field of development while others are more general-purpose programming languages.
- JavaScript is primarily used in web development and is usually the first programming language learned by beginner web developers. JavaScript can

also be used for mobile and game development. Python can be used in a variety of areas like data analysis, machine learning, and web development.

Advantages of Programming

- It enhances problem-solving skills.
- Using programming, we can solve complex problems.
- It is also learning with fun.
- It can perform multiple tasks can be bundled into one module.
- It saves time and effort.

Disadvantages of Programming

- Knowledge of computer is mandatory.
- Logical thinking should be strong.

Types of programming languages

While hundreds of programming languages (if not thousands) exist, and there are many ways to classify them, they typically can be grouped into five main categories:

1. Procedural programming languages
2. Functional programming languages
3. Object-oriented programming languages
4. Scripting languages
5. Logic programming languages

1. Procedural programming languages

A procedural language follows a sequence of statements or commands in order to achieve a desired output. Each series of steps is called a procedure, and a program written in one of these languages will have one or more procedures within it. Common examples of procedural languages include: C and C++, Java, Pascal, BASIC

2. Functional programming languages

Rather than focusing on the execution of statements, functional languages focus on the output of mathematical functions and evaluations. Each function—a reusable module of code—performs a specific task and returns a result. The result will vary depending on what data you input into the function. Some popular functional programming languages include: Scala, Erlang, Haskell, Elixir, F#

3. Object-oriented programming languages

This type of language treats a program as a group of objects composed of data and program elements, known as attributes and methods. Objects can be reused within a program or in other programs. This makes it a popular language type for complex programs, as code is easier to reuse and scale. Some common object-oriented programming (OOP) languages include: Java, Python, PHP, C++, Ruby

4. Scripting languages

Programmers use scripting languages to automate repetitive tasks, manage dynamic web content, or support processes in larger applications. Some common scripting languages include: PHP, Ruby, Python, bash, Perl, Node.js

5. Logic programming languages

Instead of telling a computer what to do, a logic programming language expresses a series of facts and rules to instruct the computer on how to make decisions. Some examples of logic languages include: Prolog, Absys, Datalog, Alma-0

Other ways to classify programming languages

Front-end vs. back-end languages

- **Front-end languages** are primarily concerned with the 'user' aspect of the software. The front-end deals with all of the text, colors, buttons, images, and navigation that the user will face when navigating your website or application. Anyone with a background in graphic design or art may be more inspired to begin learning one of the front-end languages.
- Some examples of front-end programming languages include:
 - HTML, CSS, JavaScript, React
- **Back-end languages** deal with storage and manipulation of the server side of software. This is the part of the software that the user does not directly come into contact with but supports their experience behind the scenes. This includes data architecture, scripting, and communication between applications and underlying databases.
- Anyone with experience in mathematics or engineering may find more interest in back-end development. Some examples of back-end programming languages include:
 - JavaScript, PHP, Java, Python, Ruby, C#

- A full-stack developer combines their knowledge of both front- and back-end languages, along with other technical skills and expertise, to work on any part of the development process.

High-level vs. low-level languages

- The biggest factor that differentiates high- and low-level programming languages is whether the language is meant to be easily understood by a human programmer or a computer. **Low-level languages** are machine-friendly, which makes them highly efficient in terms of memory usage but difficult to understand without the help of an assembler. Since they're not very people-friendly, they're also not widely used anymore. Examples include machine code and assembly languages.
- **High-level languages**, on the other hand, are less memory efficient but much more human friendly. This makes them easier to write, understand, maintain, and debug. Most popular programming languages in use today are considered high-level languages.

Interpreted vs. compiled languages

- The distinction between interpreted and compiled languages has to do with how they convert high-level code and make it readable by a computer. With **interpreted languages**, code goes through a program called an interpreter, which reads and executes the code line by line. This tends to make these languages more flexible and platform independent.
- Examples of interpreted languages include: Python, JavaScript, PHP, Ruby
- **Compiled languages** go through a build step where the entire program is converted into machine code. This makes it faster to execute, but it also means that you have to compile or "build" the program again anytime you need to make a change.
- Examples of compiled languages include: C, C++, and C#, Rust, Erlang

[4] What is Python?

- Python is a simple, general purpose, high level, and object-oriented programming language. **Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.** It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.
- Python is an easy-to-learn yet powerful and versatile scripting language, which makes it attractive for Application Development. With its interpreted nature, Python's syntax and dynamic typing make it an ideal language for scripting and rapid application development.
- Python supports multiple programming patterns, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. It is a multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.
- We don't need to use data types to declare variable because it is dynamically typed, so we can write `a=10` to assign an integer value in an integer variable.
- Python makes development and debugging fast because no compilation step is included in Python development, and the edit-test-debug cycle is very fast.
- Python has many web-based assets, open-source projects, and a vibrant community. Learning the language, working together on projects, and contributing to the Python ecosystem are all made very easy for developers.
- Because of its straightforward language framework, Python is easier to understand and write code in. This makes it a fantastic programming language for novices. Additionally, it assists seasoned programmers in writing clearer, error-free code.
- Python is an open-source, cost-free programming language. It is utilized in several sectors and disciplines as a result.
- In Python, code readability and maintainability are important. As a result, even if the code was developed by someone else, it is easy to understand and adapt by some other developer.
- Python has many third-party libraries that can be used to make its functionality easier. These libraries cover many domains, for example, web development, scientific computing, data analysis, and more.

Python Basic Syntax

- There is no use of curly braces or semicolon in Python programming language. It is English-like language. But Python uses the indentation to define a block of code. Indentation is nothing but adding whitespace before the statement when it is needed. Instead of Semicolon as used in other languages, Python ends its statements with a NewLine character.

- Python is a case-sensitive language, which means that uppercase and lowercase letters are treated differently. For example, 'name' and 'Name' are two different variables in Python.
- In Python, comments can be added using the '#' symbol. Any text written after the '#' symbol is considered a comment and is ignored by the interpreter. This trick is useful for adding notes to the code or temporarily disabling a code block. It also helps in understanding the code better by some other developers.
- 'If', 'otherwise', 'for', 'while', 'try', 'except', and 'finally' are a few reserved keywords in Python that cannot be used as variable names. These terms are used in the language for particular reasons and have fixed meanings. If you use these keywords, your code may include errors, or the interpreter may reject them as potential new Variables.

Features of Python

Python provides many useful features to the programmer. These features make it the most popular and widely used language. We have listed below few-essential features of Python.

- Easy to use and Learn: Python has a simple and easy-to-understand syntax, unlike traditional languages like C, C++, Java, etc., making it easy for beginners to learn.
- Expressive Language: It allows programmers to express complex concepts in just a few lines of code or reduces Developer's Time.
- Interpreted Language: Python does not require compilation, allowing rapid development and testing. It uses Interpreter instead of Compiler.
- Object-Oriented Language: It supports object-oriented programming, making writing reusable and modular code easy.
- Open Source Language: Python is open source and free to use, distribute and modify.
- Extensible: Python can be extended with modules written in C, C++, or other languages.
- Learn Standard Library: Python's standard library contains many modules and functions that can be used for various tasks, such as string manipulation, web programming, and more.
- GUI Programming Support: Python provides several GUI frameworks, such as Tkinter and PyQt, allowing developers to create desktop applications easily.
- Integrated: Python can easily integrate with other languages and technologies, such as C/C++, Java, and . NET.

- Embeddable: Python code can be embedded into other applications as a scripting language.
- Dynamic Memory Allocation: Python automatically manages memory allocation, making it easier for developers to write complex programs without worrying about memory management.
- Wide Range of Libraries and Frameworks: Python has a vast collection of libraries and frameworks, such as NumPy, Pandas, Django, and Flask, that can be used to solve a wide range of problems.
- Versatility: Python is a universal language in various domains such as web development, machine learning, data analysis, scientific computing, and more.
- Large Community: Python has a vast and active community of developers contributing to its development and offering support. This makes it easy for beginners to get help and learn from experienced developers.
- Career Opportunities: Python is a highly popular language in the job market. Learning Python can open up several career opportunities in data science, artificial intelligence, web development, and more.
- High Demand: With the growing demand for automation and digital transformation, the need for Python developers is rising. Many industries seek skilled Python developers to help build their digital infrastructure.
- Increased Productivity: Python has a simple syntax and powerful libraries that can help developers write code faster and more efficiently. This can increase productivity and save time for developers and organizations.
- Big Data and Machine Learning: Python has become the go-to language for big data and machine learning. Python has become popular among data scientists and machine learning engineers with libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and more.

Where is Python used?

Python is a general-purpose, popular programming language, and it is used in almost every technical field. The various areas of Python use are given below.

- Data Science: Data Science is a vast field, and Python is an important language for this field because of its simplicity, ease of use, and availability of powerful data analysis and visualization libraries like NumPy, Pandas, and Matplotlib.

- Desktop Applications: PyQt and Tkinter are useful libraries that can be used in GUI - Graphical User Interface-based Desktop Applications. There are better languages for this field, but it can be used with other languages for making Applications.
- Console-based Applications: Python is also commonly used to create command-line or console-based applications because of its ease of use and support for advanced features such as input/output redirection and piping.
- Mobile Applications: While Python is not commonly used for creating mobile applications, it can still be combined with frameworks like Kivy or BeeWare to create cross-platform mobile applications.
- Software Development: Python is considered one of the best software-making languages. Python is easily compatible with both from Small Scale to Large Scale software.
- Artificial Intelligence: AI is an emerging Technology, and Python is a perfect language for artificial intelligence and machine learning because of the availability of powerful libraries such as TensorFlow, Keras, and PyTorch.
- Web Applications: Python is commonly used in web development on the backend with frameworks like Django and Flask and on the front end with tools like JavaScript and HTML.
- Enterprise Applications: Python can be used to develop large-scale enterprise applications with features such as distributed computing, networking, and parallel processing.
- 3D CAD Applications: Python can be used for 3D computer-aided design (CAD) applications through libraries such as Blender.
- Machine Learning: Python is widely used for machine learning due to its simplicity, ease of use, and availability of powerful machine learning libraries.
- Computer Vision or Image Processing Applications: Python can be used for computer vision and image processing applications through powerful libraries such as OpenCV and Scikit-image.
- Speech Recognition: Python can be used for speech recognition applications through libraries such as SpeechRecognition and PyAudio.
- Scientific computing: Libraries like NumPy, SciPy, and Pandas provide advanced numerical computing capabilities for tasks like data analysis, machine learning, and more.

- Education: Python's easy-to-learn syntax and availability of many resources make it an ideal language for teaching programming to beginners.
- Testing: Python is used for writing automated tests, providing frameworks like unit tests and pytest that help write test cases and generate reports.
- Gaming: Python has libraries like Pygame, which provide a platform for developing games using Python.
- IoT: Python is used in IoT for developing scripts and applications for devices like Raspberry Pi, Arduino, and others.
- Networking: Python is used in networking for developing scripts and applications for network automation, monitoring, and management.
- DevOps: Python is widely used in DevOps for automation and scripting of infrastructure management, configuration management, and deployment processes.
- Finance: Python has libraries like Pandas, Scikit-learn, and Statsmodels for financial modeling and analysis.
- Audio and Music: Python has libraries like Pyaudio, which is used for audio processing, synthesis, and analysis, and Music21, which is used for music analysis and generation.
- Writing scripts: Python is used for writing utility scripts to automate tasks like file operations, web scraping, and data processing.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.