MODULE-1[SDLC]

QUESTION-1:

- What is software?

- Software is a set of instruction, data, or a program used to operate a computer and execute specific tasks. In a simpler terms, software tells a computer how to function. It’s a generic term used to refer to application, scripts and programs that run on device such as PCs, mobile phones, tablets and other smart devices. Software contrasts with hardware which is the physical aspect of a computer that perform the work.
- Without software most computer would be useless. For example a web browser is a software application that allows user to access the internet.
- Without the web browser software, reading this page on webopedia wouldn’t be possible.

- An operating system (OS) is a software program that serves as the interface between other application and the hardware on the computer or a mobile. TCP/IP is built into all major operating system to allow computers to communicate over long distance networks. Without the OS or the protocols built into it, it wouldn't be possible to access a web browser.
- The majority of software is written in high-level programming languages due to the language being closer to natural human language as opposed to machine language. The high-level language is then translated into low level machine code using a compiler or interpreter for the computer to understand. Software can also be written in a low-level assembly language, but it is less common.

HISTORY OF SOFTWARE

- Ada Lovelace wrote the first known computer program in 1843 for the Analytical Engine. The Analytical Engine was designed by Charles Babbage in 1837 and was the concept for the first general mechanical computer. The program, however, remained theoretical as the Analytical Engine was never physically constructed. The first modern theory of software was proposed by Alan Turing in his 1935 essay, “Computable numbers with an application to the Entscheidung’s problem (decision problem)The first time a stored-program computer held a piece of software in electronic memory and executed it successfully was on June 21, 1948.
- Another major innovation in the history of software development was the emergence of open-source software in the 1990s. The Linux kernel was released in 1991, and interest in open-source software skyrocketed after the 1998

TYPES OF SOFTWARE

- APPLICATION SOFTWARE
- SYSTEM SOFTWARE
- PROGRAMMING SOFTWARE
- DRIVER SOFTWARE

❖ APPLLICATION SOFTWARE :

❖ Application software is software that helps an end user complete task such as doing research, taking notes, setting an alarm, designing graphics, or keeping an account log. Application software lies above the system software and is different from system software in that it's designed for the end use and is specific in its functionality. This type of software is sometimes referred to as non-essential software because it's installed and operated based on the user's needs.

❖ SYSTEM SOFTWARE :

- System software helps the user, hardware, and application software interact and function with each other. System software acts as a mediator or middle layer between the user and the hardware. It’s essential in managing the whole computer system - when a computer is first turned on, it’s the system software that is initially loaded into memory. Unlike application software, system software isn’t used by end users. Instead, it runs in the background of a device. The most well-known example of system software is the OS, which manages all other programs in a computer.

- **Basic input/output system (BIOS)**: the built-in firmware that determines what a computer can do without accessing programs from a disk.
- **Boot**: loads the OS into the computer’s main memory or RAM.
- **Assembler**: Takes basic instructions and converts them into a pattern of bits that the processor can use to perform basic operations.

❖ PROGRAMMING SOFTWARE :

- Classified as a type of system software, programming software isn’t used by the end user. It’s used by programmers who are writing code. Programming software is a program that is used to write, develop, test, and debug other software, including application and system software. These programs serve as a sort of translator. It takes programming languages such as Python or C++ and translates it into something a computer will understand, known as machine language code.

- Assigns data storage
- Enlists source code as well as program details
- Offers diagnostic reports
- Recifties system errors during runtime

❖ DRIVER SOFTWARE :

❖ Also classified as a type of system software, driver software operates and controls devices and peripherals plugged into a computer, enabling a device to perform the designated task. Hardware devices that need a driver to connect to a system include displays, sound cards, printer, mice, and hard disks. Since there are numerous types of devices, drivers allow software systems to communicate through a standardized language. An OS typically comes with built-in drivers for a mouse, keyboard, and printer by default, so third-party installations aren’t required. For advanced devices, the driver may need to be installed externally. If multiple OS are used, such as Linux, Windows, or Mac, separate drivers need to be maintained for each. Examples of drivers include:

- BIOS driver
- Display driver
- Motherboard driver
- ROM driver

QUESTION-2:

What are the types of Applications?

- Application software is software that helps an end user complete task such as doing research, taking notes, setting an alarm, designing graphics, or keeping an account log. Application software lies above the system software and is different from system software in that it’s designed for the end use and is specific in its functionality. This type of software is sometimes referred to as non-essential software because it’s installed and operated based on the user’s needs.

- Web browsers like Firefox, and Google Chrome, as well as Microsoft Word and Excel, are examples of application software that is used on a personal computer or laptop.

- It also includes smartphone apps such as WhatsApp and Telegram, as well as games such as Candy Crush Saga and Ludo. There are also app versions of popular services that people rely on every day, such as weather or transportation information, as well as apps that connect end users with their businesses.

- Application software can also be classified depending on how much it costs and how easily it can be accessed. Here are some examples of application software:

1. Word Processing Software

- Word Processing refers to the act of using a personal computer (PC) or laptop to create, edit, save and print documents which can be performed only with specialized software known as a Word Processor.

- One example of a Word Processor is Microsoft Word which is widely used by all professionals.

2. Spreadsheet Software

- Spreadsheet software is a type of computer program that enables a user to perform numerical functions and explore numbers through an automated version of an accounting worksheet. Best example of spreadsheet software is Microsoft Excel.

3. Presentation Software

- Presentation software also commonly known as presentation graphics is a particular category of application program used to construct sequences of words and a series of pictures that tell a story or help support a speech or public presentation of any type of information or a launch of new products or services.

4. Multimedia Software

❖ Multimedia software can be described as the combination of text, audio, images, animation, or video to produce a wide scope of interactive content for both professional and personal use. You can easily learn about media players, file formats, and how to operate audio and video software on the whole

5. Web Browsers

❖ A web browser can take you all over the internet. It retrieves data from other parts of the web and shows it on your desktop or mobile device for your viewing. The data is transmitted using the Hypertext Transfer Protocol, which describes how text, images, and video are shared on the World Wide Web.

6. Educational Software

❖ Educational software refers to any computer software designed solely for educational reasons.

- It includes a wide range of software, including language learning software, classroom management software (CMS), and reference software for students and other professionals.

7. Graphics Software

- Graphics software can rework with bitmap and/or vector graphics and can be utilized to create label templates. Graphics software generally includes Canva, Adobe Illustrator, Photoshop, InDesign, CorelDraw, Inkscape, Microsoft Paint, and Paint.Net.

8. Freeware

- Freeware is typically marketed for profit but might be allocated specifically for a business or commercial purpose with the aim to expand the market share of any newly launched premium product. Some of the widespread examples of closed-source freeware include Adobe Reader, Free Studio, and Skype.

9. Shareware

- Shareware is software that is supplied for free on a trial basis in order for the user to test or use the programme for a specific amount of days with the understanding that the user may need or want to pay for it later if they are satisfied with the product usage. Some software manufacturers provide a shareware edition of their product with an expiration date built in, such that after 30 days, the user or customer will no longer be able to access the application for further use.

10. Simulation Software

- Simulation software authorizes engineers to evaluate, optimize, and compare product designs with other similar software by modeling real-world events in a computer-generated environment.

11. Open Source

- Open source software is a specific code designed to be publicly accessible so that anyone can see,modify, and distribute the code as they see which fits the purpose. It is designed in a decentralized and coordinated way, depending on peer assessment and community production.

12. Closed Source

❖ Closed source software is where the source code is not freely accessible. It is developed and delivered to the customer as a fully compiled, executable set of files. The developer often provides aid to users after purchase and ensures that the software works as foreseen by the creator.

QUESTION-3:

❖ What is programming?

❖ Programming refers to a technological process for telling a computer which tasks to perform in order to solve problems. You can think of programming as a collaboration between humans and computers, in which humans create instructions for a computer to follow (code) in a language computers can understand.

❖ Programming enables so many things in our lives. Here are some examples:

❖ When you browse a website to find information, contact a service provider, or make a purchase, programming allows you to interact with the site's on-page elements, such as sign-up or purchase buttons, contact forms, and drop-down menus.

❖ The programming behind a mobile app can make it possible for you to order food, book a rideshare service, track your fitness, access media, and more with ease.

- Programming helps businesses operate more efficiently through different software for file storage and automation and video conferencing tools to connect people globally, among other things.
- Space exploration is made possible through programming.

Most used programming languages

- Different programming languages enable programmers to write code that computers understand. According to a survey by Statista, the top five programming languages that developers use are:

- JavaScript
- SQL
- Python
- TypeScript
- HTML/CSS

QUESTION-4:

❖ What is python?

- ❖ Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. It is high level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapi application development, as well as for use as a scripting or glue language to connect existing components together. Python is simple, easy to learn syntax emphasizes readability and therefor reduces the cost of program maintenance. Python supports moduels and packages, which encourages program modularity and code reuse. The python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.
- ❖ Often, programmers fall in love with python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug

cycle is incredibly fast. Debugging python program is easy a bug or a bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn’t catch the exception, the interpreter prints a stack trace. A souce level debugger allows inspection of local and global variables, evaluation of arbitrary expression, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in python itself, testifying to python’s introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.