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. The two main categories of software are application software and system software. An application is software that fulfills a specific need or performs tasks. System software is designed to run a computer's hardware and provides a platform for applications to run on top of.

Other types of software include programming software, which provides the programming tools software developers need; middleware, which sits between system software and applications; and driver software, which operates computer devices and peripherals.

Early software was written for specific computers and sold with the hardware it ran on. In the 1980s, software began to be sold on floppy disks, and later on CDs and DVDs. Today, most software is purchased and directly downloaded over the internet. Software can be found on vendor websites or application service provider websites.

- ## what is software engineering?

Software engineering is defined as a process of analyzing user requirements and then designing, building, and testing software application which will satisfy those requirements.

IEEE, in its standard 610.12-1990, defines software engineering as the application of a systematic, disciplined, which is a computable approach for the development, operation, and maintenance of software.

Fritz Bauer defined it as ‘the establishment and used standard engineering principles. It helps you to obtain, economically, software which is reliable and works efficiently on the real machines’.

Boehm defines software engineering, which involves, ‘the practical application of scientific knowledge to the creative design and building of computer programs. It also includes associated documentation needed for developing, operating, and maintaining them.’

2. Explain types of software

Application software. The most common type of software, application software is a computer software package that performs a specific function for a user, or in some cases, for another application. An application can be self-contained, or it can be a group of programs that run the application for the user.

System software. These software programs are designed to run a computer's application programs and hardware. System software coordinates the activities and functions of the hardware and software. In addition, it controls the operations of the computer hardware and provides an environment or platform for all the other types of software to work in. The OS is the best example of system software; it manages all the other computer programs. computer language translators and system. **Driver software.** Also known as device drivers, this software is often considered a type of system software. Device drivers control the devices and peripherals connected to a computer, enabling them to perform their specific tasks. Every device that is connected to a computer needs at least one device driver to function. Examples include software that comes with any nonstandard hardware, including special game controllers, as well as the software that enables standard hardware, such as USB storage devices, keyboards, headphones and printers.

Middleware. The term *middleware* describes software that mediates between application and system software or between two different kinds of application software. For example, middleware enables Microsoft Windows to talk to Excel and Word. It is also used to send a remote work request from an application in a computer that has one kind of OS, to an application in a computer with a different OS. It also enables newer applications to work with legacy ones.

Programming software. Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

2. What is SDLC? Explain each phase of SDLC

SDLC (Software Development Life Cycle) is used in Every Software Development Company because it is the root of the Development Cycle, if that model would not exist in the world, firstly no software can build secondly if any how it would be made, it’s not going to succeed it has no use, because of no maintenance, but Luckily SDLC model exist in Tech world But why we need it Actually!

There are several reasons why organizations use the Software Development Life Cycle (SDLC) when developing software applications:

To provide a **structured and organized approach** to software development: The SDLC provides a framework for managing the software development process, which helps to ensure that all necessary steps are taken and that the final product meets the requirements.

1.To ensure that the software is of high quality: The SDLC includes testing and quality assurance phases, which help to ensure that the software is free of bugs and that it meets the requirements.

2.To manage risks and costs: The SDLC helps organizations to identify and manage risks early in the development process, which can help to reduce costs and minimize the impact of any issues that do arise.

3.To improve communication and collaboration: The SDLC helps to ensure that all stakeholders, including customers, end-users, and developers, are involved in the development process and that their needs are taken into account.

4.To improve efficiency and productivity: The SDLC helps organizations to optimize the use of resources and to streamline the development process, which can improve efficiency and productivity.

5.To increase the likelihood of a successful project outcome: Following a well-defined SDLC process can greatly increase the chances of success of the project, as the process guides the team towards the goal in a systematic and efficient way.

Overall, the **SDLC is a valuable tool** for organizations to use when developing software applications, as it helps **to ensure that the final product is of high quality, meets the requirements**, and is delivered on time and within budget.

- **Now we show SDLC phase:**

1.Requirements gathering and analysis: This phase involves gathering information about the software requirements from stakeholders, such as customers, end-users, and business analysts.

2. Design: In this phase, the software design is created, which includes the overall architecture of the software, data structures, and interfaces. It has two steps:

High-level design (HLD): It gives the architecture of software products.

Low-level design (LLD): It describes how each and every feature in the product should work and every component.

3. Implementation or coding: The design is then implemented in code, usually in several iterations, and this phase is also called as Development.

things you need to know about this phase:

This is the longest phase in SDLC model.

This phase consists of Front end + Middleware + Back-end.

In front-end: Development of coding is done even SEO settings are done.

In Middleware: They connect both the front end and back end.

In the back-end: A database is created.

4. Testing: The software is thoroughly tested to ensure that it meets the requirements and works correctly.

5. Deployment: After successful testing, The software is deployed to a production environment and made available to end-users.

6. Maintenance: This phase includes ongoing support, bug fixes, and updates to the software.

There are **different methodologies** that organizations can use to implement the SDLC, such as **Waterfall, Agile, Scrum, V-Model** and **DevOps.**

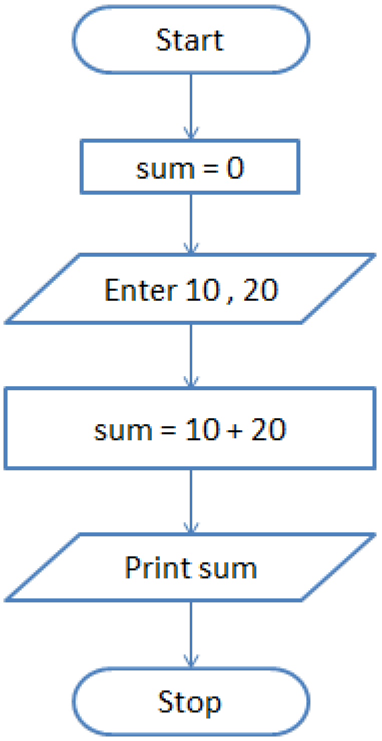
What is DFD? Create a DFD diagram on Flipkart.

DFD is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart.

It is a graphical tool, useful for communicating with users, managers and other personnel. it is useful for analysing existing as well as proposed system.

What is flow chart? Create a flow chart of addition of two numbers.

A flowchart is a diagram that depicts a process, system or computer algorithm. They are widely used in multiple fields to document, study, plan, improve and communicate often complex processes in clear, easy-to-understand diagrams. Flowcharts, sometimes spelled as flow charts, use rectangles, ovals, diamonds and potentially numerous other shapes to define the type of step, along with connecting arrows to define flow and sequence. They can range from simple, hand-drawn charts to comprehensive computer-drawn diagrams depicting multiple steps and routes. If we consider all the various forms of flowcharts, they are one of the most common diagrams on the planet, used by both technical and non-technical people in numerous fields. Flowcharts are sometimes called by more specialized names such as Process Flowchart, Process Map, Functional Flowchart, Business Process Mapping, Business Process Modeling and Notation (BPMN), or Process Flow Diagram (PFD). They are related to other popular diagrams, such as Data Flow Diagrams (DFDs) and Unified Modeling Language (UML) Activity Diagrams.



What is use case diagram? Create a use case on bill payment on paytm.

A **use case diagram** is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures.