**1.** **What is Software? What is Software Engineering?**

* **Software :-**

**Software** is a collection of instructions, data, or computer programs that are used to run machines and carry out particular activities. It is the antithesis of hardware, which refers to a computer’s external components. A device’s running programs, scripts, and applications are collectively referred to as “software” in this context.

**Software** is a program or set of programs containing instructions that provide desired functionality. Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.

* **Software Engineering :-**

1. Software engineering includes a variety of techniques, tools, and methodologies, including requirements analysis, design, testing, and maintenance.
2. It is a rapidly evolving field, and new tools and technologies are constantly being developed to improve the software development process.
3. By following the principles of software engineering and using the appropriate tools and methodologies, software developers can create high-quality, reliable, and maintainable software that meets the needs of its users.
4. Software Engineering is mainly used for large projects based on software systems rather than single programs or applications.
5. The main goal of Software Engineering is to develop software applications for improving quality,  budget, and time efficiency.
6. Software Engineering ensures that the software that has to be built should be consistent, correct, also on budget, on time, and within the required requirements.

**2. Explain types of Software?**

* **The types of Software :-**



1. **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. Examples of modern applications include office suites, graphics software, databases and database management programs, web browsers, word processors, software development tools, image editors and communication platforms.
2. **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. Other examples of system software include the firmware, computer language translators and system utilities.

1. **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.
2. **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.
3. **Programming software:** Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

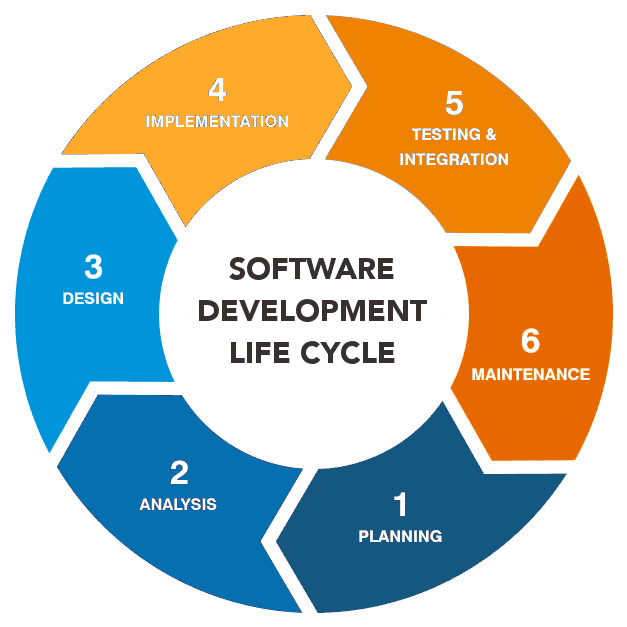
**3. What is SDLC? Explain each phase of SDLC?**

* **What is SDLC :-**

Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step.

The goal of the SDLC life cycle model is to deliver high-quality, maintainable software that meets the user’s requirements. SDLC in software engineering models outlines the plan for each stage so that each stage of the software development model can perform its task efficiently to deliver the software at a low cost within a given time frame that meets users’ requirements.

* **Each phase of SDLC :-**



1. [**Requirements gathering and analysis:**](https://www.geeksforgeeks.org/requirements-gathering-introduction-processes-benefits-and-tools/)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.

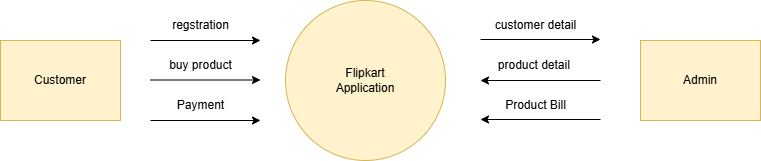
There are **different**methodologies that organizations can use to implement the SDLC, such as**Waterfall, Agile, Scrum, V-Model**and**DevOps.**

Summer-time is here and so is the time to skill-up! More than 5,000 learners have now completed their journey from **basics of DSA to advanced level development programs** such as Full-Stack, Backend Development, Data Science.

**4. What is DFD? Create a DFD diagram on Flipkart.**

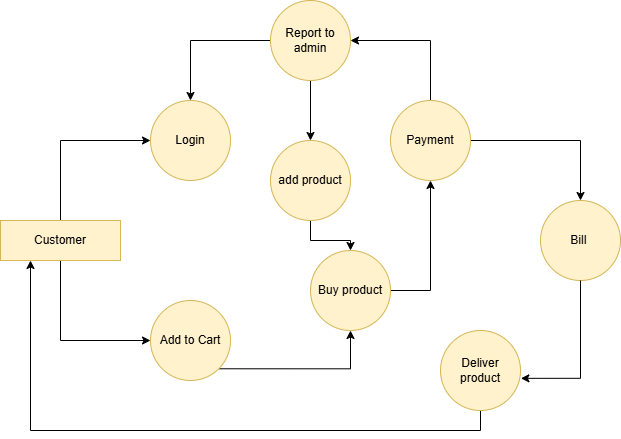
* **DFD (Data Flow Diagram) :-** **DFD** is the abbreviation for **Data Flow Diagram**. The flow of data in a system or process is represented by a Data Flow Diagram (DFD). It also gives insight into the inputs and outputs of each entity and the process itself. Data Flow Diagram (DFD) does not have a control flow and no loops or decision rules are present.
* **0 Level DFD :-**

Level 0 is the highest-level Data Flow Diagram (DFD), which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes.



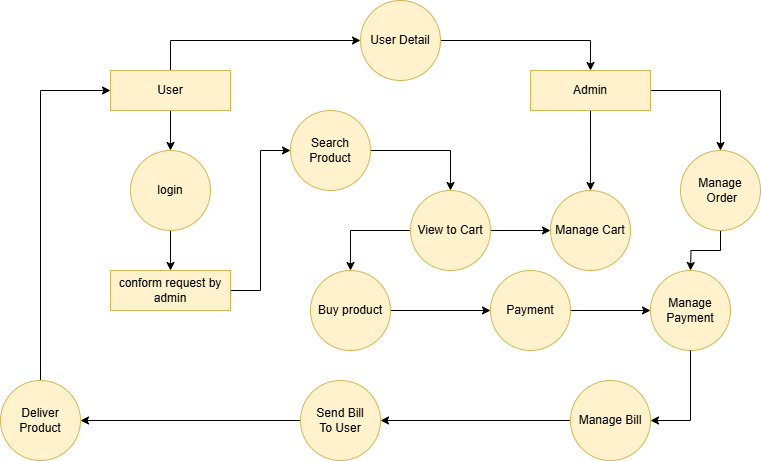
* **1 level DFD :-**

1-Level provides a more detailed view of the system by breaking down the major processes identified in the level 0 Data Flow Diagram (DFD) into sub-processes. Each sub-process is depicted as a separate process on the level 1 Data Flow Diagram (DFD). The data flows and data stores associated with each sub-process are also shown.



* **2 Level DFD :-**

2-Level provides an even more detailed view of the system by breaking down the sub-processes identified in the level 1 Data Flow Diagram (DFD) into further sub-processes. Each sub-process is depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process are also shown.



**5. What is Flow chart? Create a flowchart to make addition of two numbers.**

* **Flowchart** :- Flowcharts are nothing but the graphical representation of the data or the algorithm for a better understanding of the code visually.

It displays step-by-step solutions to a problem, algorithm, or process.

* **Algorithm** :- A set of rules that must be followed when solving a particular problem

**Program :-**

#include<stdio.h>

int main ()

{

int n1, n2, answer;

Print f ("enter 2 values");

Scan f ("%d %d", &n1, &n2);

answer = n1+n2;

print f ("%d is answer", answer);

return 0;

}

**Algorithm** :-

1)start

2)declare n1, n2, answer

3)display message enter 2 values

4)read values of n1 and n2

5)answer->n1+n2

6)display answer

7)stop

* **Flowchart to make addition of two numbers :-**



**Flow Chart**

**6. What is Use case Diagram? Create a use-case on bill payment on Paytm.**

* **Use-Case Diagram** :- use-case diagram illustrate and define the context and requirement of either an entire system or the important parts of the system. You can model a complex system with a single use case diagram or create many use case diagrams to model the components of the system.
* **Use-case Diagram Bill Payment on Paytm :-**

