

MODULE: 1 (SDLC)

1. What is software? What is software engineering?

Ans.

- Software is a program or set of programs containing instructions that provide desired functionality. And Engineering is the process of designing and building something that serves a particular purpose and finds a cost-effective solution to problems.
- Software Engineering is mainly used for large projects based on software systems rather than single programs or applications. The main goal of software Engineering is to develop software application for improving the quality, budget and time efficiency. 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.
- There are Four main Attributes of Software Engineering:-
 - Efficiency
 - Reliability
 - Robustness
 - Maintainability

2. Explain types of software

Ans.

- **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.

- **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.
- **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 debug other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

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

Ans :- The Software Development Life Cycle (SDLC) is a structured process that enables the production of high-quality, low-cost software, in the shortest possible production time. The goal of the SDLC is to produce superior software that meets and exceeds all customer expectations and demands.

Requirements Analysis

The first step of any SDLC is to define the project's requirements. Some critical questions during this stage are:

- What is the goal of the new project?
- What is the business hoping to get from the product?
- Is the team going to write code from scratch, or are we upgrading an existing system?
- Do we have any hard deadlines?
- Do we have the necessary knowledge in-house, or will we have to outsource some part(s) of the project?

This stage requires a combined effort of business analytics, operations, leadership, development, and security teams. In some use cases, asking end users for input is also a valuable source of info.

All the data gathered in this stage goes into a **Software Requirement Specification (SRS) document**. An SRS file includes all the software, hardware, security, and network specifications for the upcoming product, but the file also contains info concerning:

- Resource allocation.
- Capacity planning.
- Project scheduling.
- Cost estimation.
- Provisioning.

The output of this step: An SRS document that defines project goals and scope, plus provides product requirements and rough project estimations (budget, resources, deadlines, etc.).

Feasibility Study

Senior business analysts perform a feasibility study to determine the software's viability. The usual approach is to focus primarily on these five factors:

- Budget constraints.
- Legal implications.
- Operational requirements.
- Available in-house skills.
- The required project timeframe.

Analysts add the findings of this phase to the existing SRS document, after which a team of decision-makers reviews the report to approve:

- Project plans and direction.
- Estimated costs.
- Projected schedules.
- The necessary resources.

The higher management either signs off on the project or asks the team to go back a step in the SDLC and come up with a new suggestion.

The output of this step: An expanded SRS document approved by higher management.

Design Plan

Once there is an approved project direction, the team starts creating a design plan that explains all major aspects of the new product, including its:

- Architecture (programming language, databases, interfaces, operating system, pre-made templates, APIs, etc.).
- Features list.
- Infrastructure requirements.
- UI design.
- Necessary security measures (e.g., SSL encryption, password protection, recommended database migrations, etc.).

The team gathers this info in **the Design Document Specification (DDS)**. A stakeholder reviews the DDS and approves a direction based on the following factors:

- Design modularity.
- Risk assessment.
- Product robustness.
- Time constraints.

Some companies decide to create a prototype during this SDLC stage. While time-consuming, prototyping is much less expensive than making radical changes after the development phase.

The output of this step: A detailed DDS that lists all the info developers require to code the product.

Software Development

The development team gets familiar with the DDS and starts working on the code. Typically, this step is the most time-consuming phase of SDLC, so we recommend using agile methodologies to speed up coding.

This phase results in operational software that meets all the requirements listed in the SRS and DDS. While the code still awaits advanced testing, the team should already put the product through basic tests (such as static code analysis and code reviews for multiple device types).

The output of this step: Source code of a testable, fully functional software.

In-Depth Software Testing

The software that comes out of the previous SDLC phase now goes through extensive testing. Companies have a wide variety of testing methods to evaluate the new product, such as:

- Code quality testing.
- Unit testing (functional tests).
- Integration testing.
- Performance testing.
- Security testing.
- Acceptance testing.
- Nonfunctional testing.

Most teams rely on automated tests to speed up this phase, but some manual examinations are also valuable (penetration tests are a good example).

If the team discovers a defect, the code goes back a step in its life cycle, and developers create a new, flaw-free version of the software. The testing stage ends when the product is stable, free of bugs, and up to quality standards defined in the previous phases.

The output of this step: A thoroughly tested version of the product ready for a production environment.

Software Deployment

The product leaves the testing phase and is ready to go into production. Some projects require the team to write user manuals or create instructional videos before the software becomes available to end users.

Ideally, the deployment phase happens automatically (typically as a part of CI/CD). Companies with lower maturity or in some highly regulated industries may require manual approvals during this SDLC stage.

Most companies deploy new software to a small percentage of users (10 to 15%) and slowly phase it into the rest of the customer base. Gradual introduction means you limit the impact on the UX if there's an overlooked issue with the product.

The output of this step: A fully functional and tested product that's available to end users.

Product Maintenance and Enhancement

Every shipped piece of software requires periodic reviews and updates based on user feedback. The most common activities during this stage are:

- Bug fixing.
- Setting up continuous monitoring.
- Upgrading the app to the newer version.
- Adding new features to the software.

Whenever a user reports a bug or the team discovers a new flaw, the product moves back through its SDLC as many steps as necessary. Some severe defects require updates in the design stage, while most problems take the app back to the development stage.

The output of this step: A fully monitored product that continuously sees improvements.

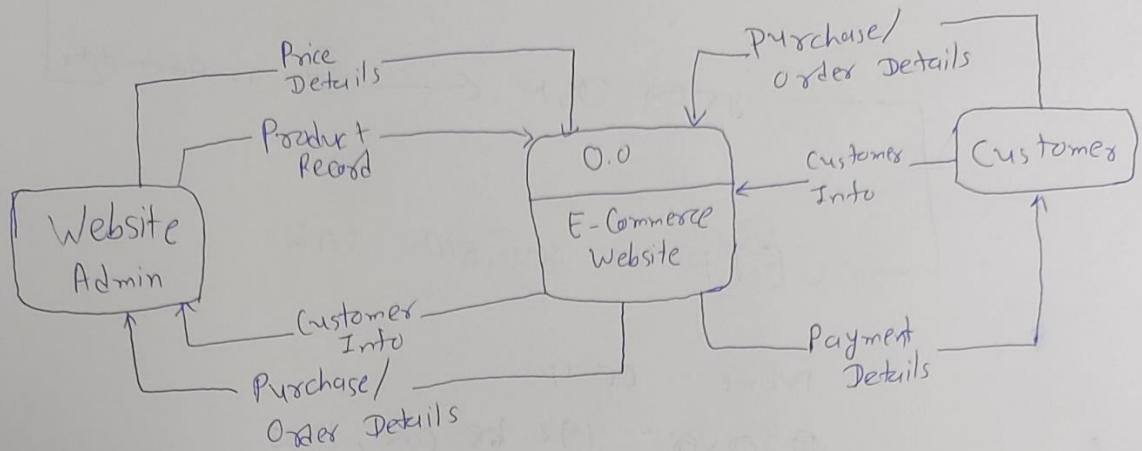
4. What is DFD? Create a DFD diagram on Flipkart.

Ans :- A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

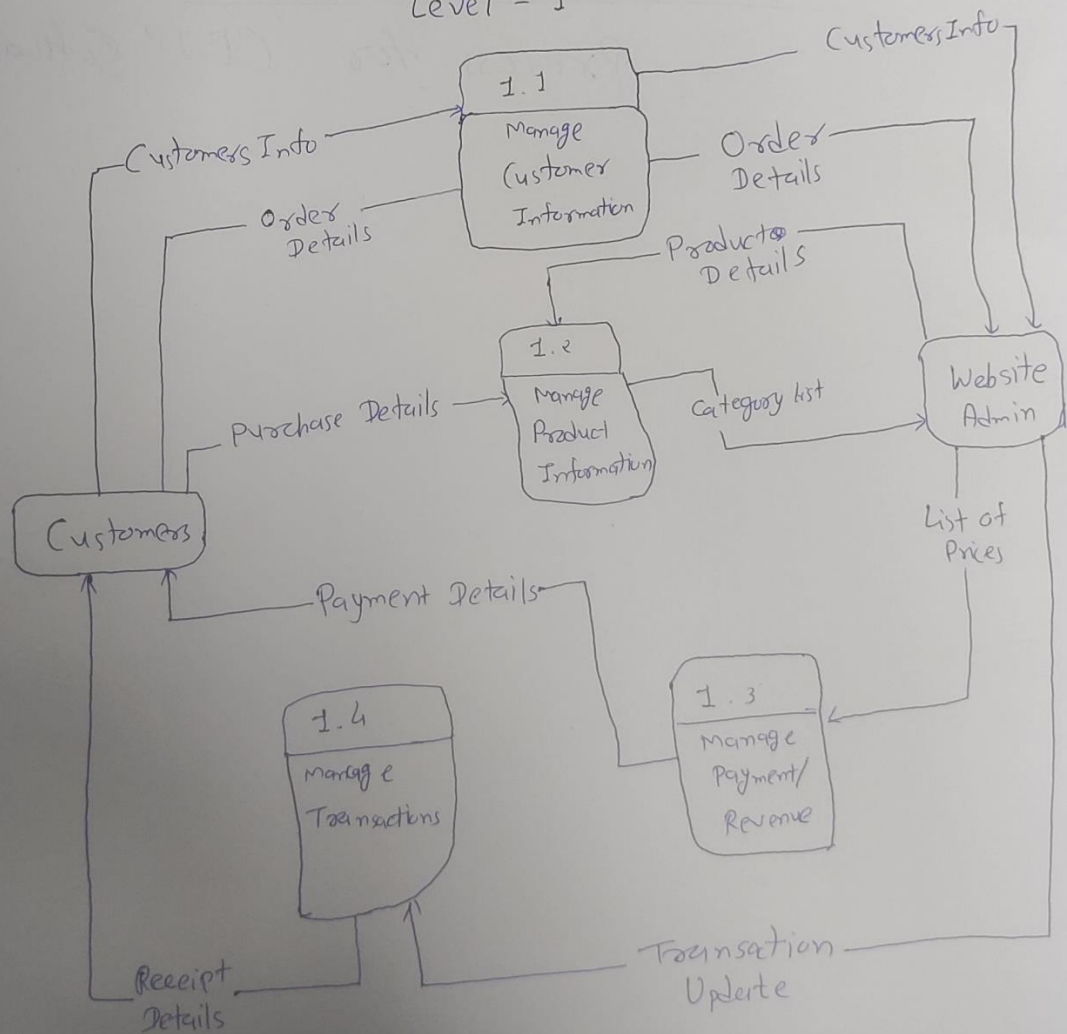
It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

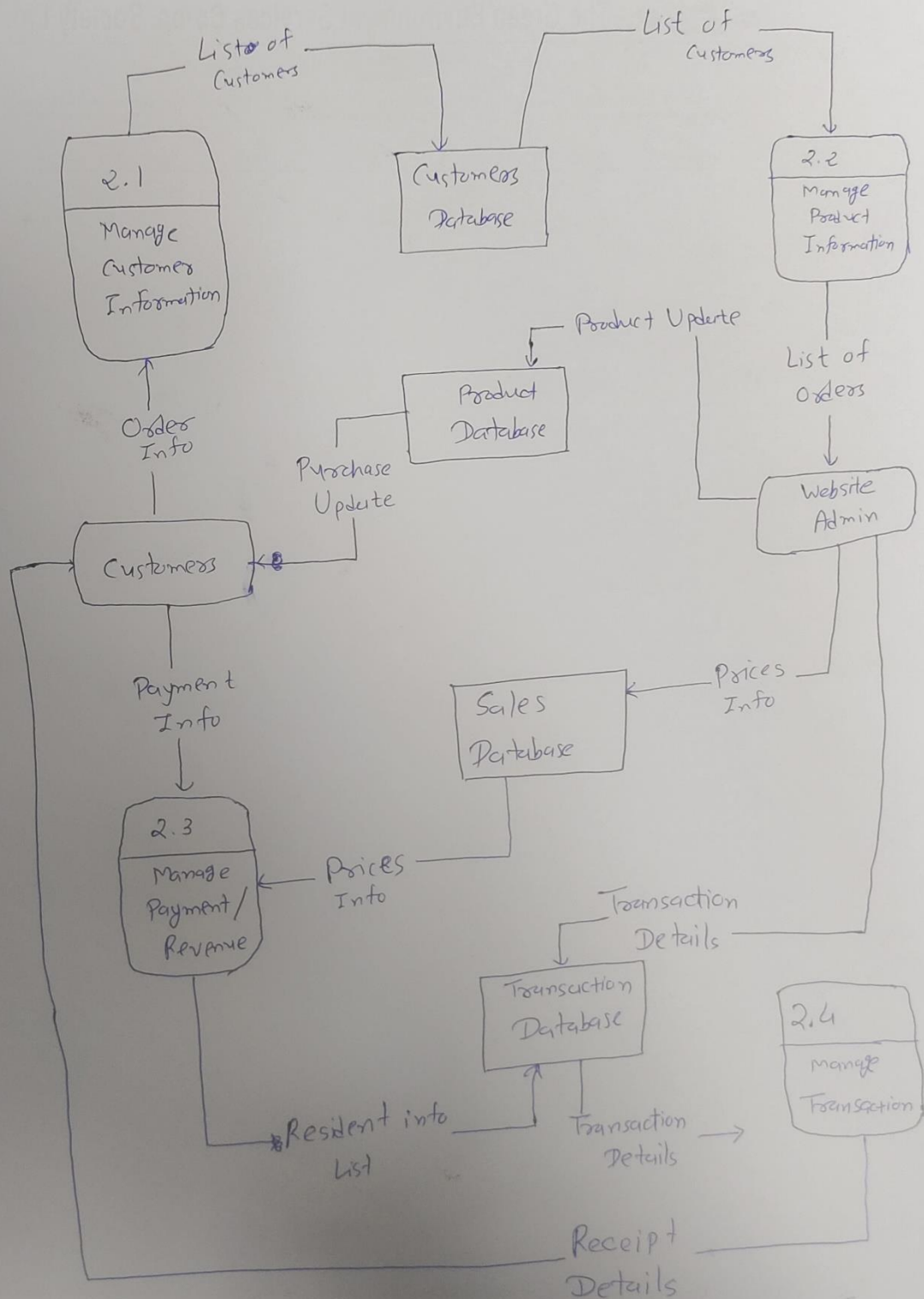
Level - 0



Level - 1



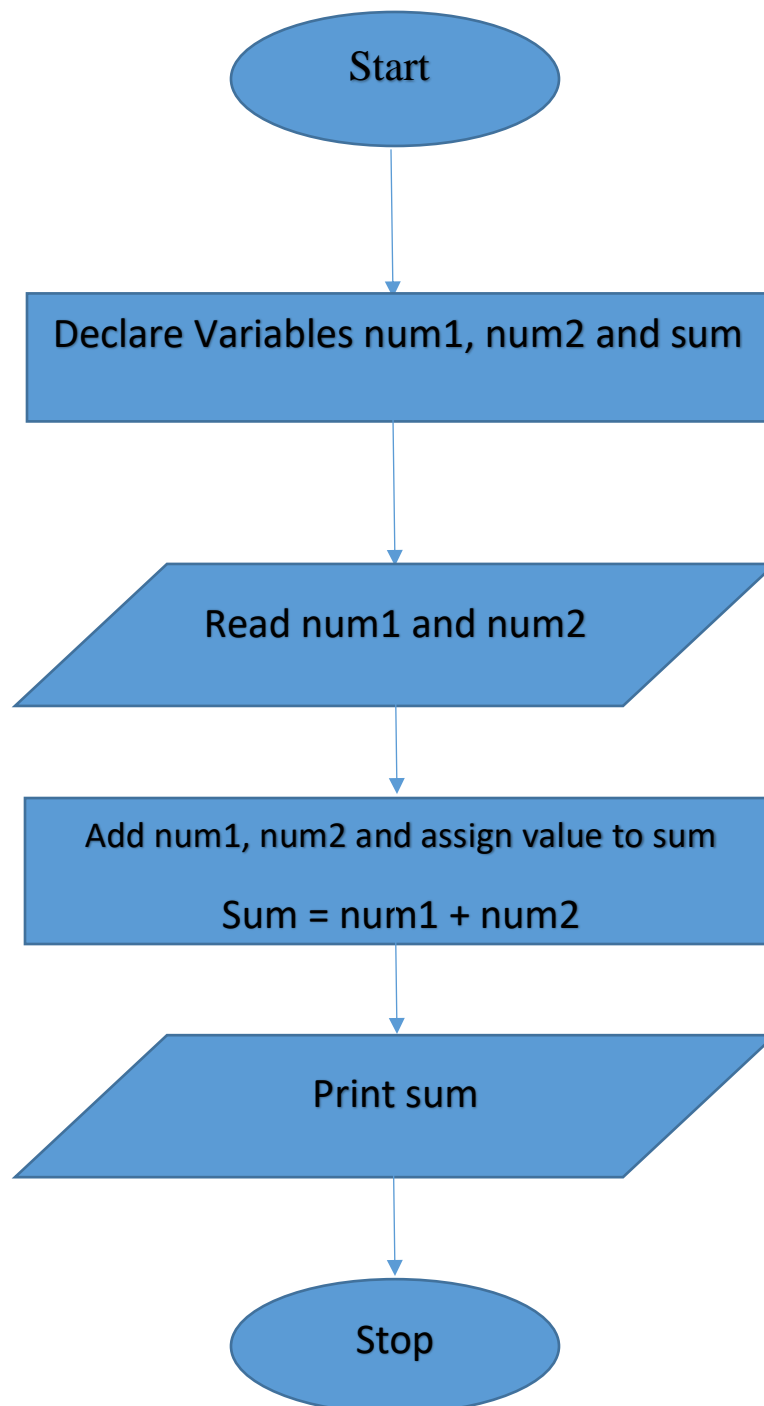
Level-2



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

Ans :- A flowchart is **a diagram that shows an overview of a program** . Flowcharts normally use standard symbols to represent the different types of instructions . These symbols are used to construct the flowchart and show the step-by-step solution to the problem. Flowcharts are sometimes known as flow diagrams.

Flow Chart



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

Ans :- Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally.

