MODULE -1

SE - Overview of IT Industry

1. **What is software?**

# “It's a set of instructions or programs that tell the computer how to perform different tasks”

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.

# **What is Software Engineering?**

**Software engineering** is the branch of computer science that deals with the **design, development, testing, and maintenance of software applications.** Software engineers apply engineering principles and knowledge of programming languages to build software solutions for end users.

Software engineers design and develop computer games, business applications, operating systems, network control systems, and middleware—to name just a few of the many career paths available.

A career as a software engineer can be both fun and challenging with opportunities to work in almost any industry, including large and small businesses, government agencies, nonprofit organizations, healthcare facilities, and more. And as technology continues to evolve, the need for software developers continues to grow. Many companies are also shifting towards hiring software engineers who work from home, allowing for **increased flexibility and more opportunities** to enter the field.

1. **Explain types of Software.**

There are several types of software based on their functionalities, which are as follows :

1. System Software:

System software is essential for the basic operation of a computer system. It provides a platform for running application software and manages computer hardware and resources. Examples include:

- Operating Systems (e.g., Windows, macOS, Linux)

- Device Drivers

- Utility Software (e.g., antivirus programs, disk cleaners, system optimizers)

2. Application Software:

Application software serves specific purposes for end-users. It is designed to perform tasks or solve problems for individuals, businesses, or organizations. Examples include:

- Word Processors (e.g., Microsoft Word, Google Docs)

- Spreadsheets (e.g., Microsoft Excel, Google Sheets)

- Presentation Software (e.g., Microsoft PowerPoint, Prezi)

- Database Management Systems (e.g., MySQL, Microsoft Access)

- Web Browsers (e.g., Google Chrome, Mozilla Firefox)

- Multimedia Software (e.g., Adobe Photoshop, VLC Media Player)

3. Programming Software:

Programming software provides tools and environments for writing, testing, and debugging software programs. These tools are used by software developers to create applications and system software. Examples include:

- Integrated Development Environments (IDEs) (e.g., Visual Studio, Eclipse, IntelliJ IDEA)

- Text Editors (e.g., Sublime Text, Atom, Vim)

- Compilers, Interpreters, and Debuggers (e.g., GCC, Python Interpreter, GDB)

4. Embedded Software:

Embedded software is specialized software that is built into hardware devices to control their functions. It is typically found in devices such as consumer electronics, automobiles, medical devices, and industrial machines. Examples include:

- Firmware for IoT devices

- Automotive control systems

- Medical device software

5. Enterprise Software:

Enterprise software is designed to address the needs of large organizations or businesses. It typically includes applications for managing various aspects of business operations, such as:

- Enterprise Resource Planning (ERP) systems

- Customer Relationship Management (CRM) software

- Supply Chain Management (SCM) software

- Human Resource Management (HRM) software

6. Utility Software:

Utility software provides additional functionality to enhance the performance or usability of a computer system. These tools are often used for maintenance, security, and optimization purposes. Examples include:

- Antivirus and Antimalware software

- Backup and Recovery software

- Disk Cleanup and Optimization tools

- Encryption software

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

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deploying and maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning, creating, testing, and deploying a Software Product.

**\*Phases of SDLC life Cycles**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2018/04/SDLC-Cycle.jpg)

**1)Requirement Gathering and Analysis:**

During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

**2) Design:**

In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

**3) Implementation and coding:**

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

**4) Testing:**

Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

**5) Deployement:**

Once the product is tested, it is deployed in the production environment or first [UAT (User Acceptance testing)](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

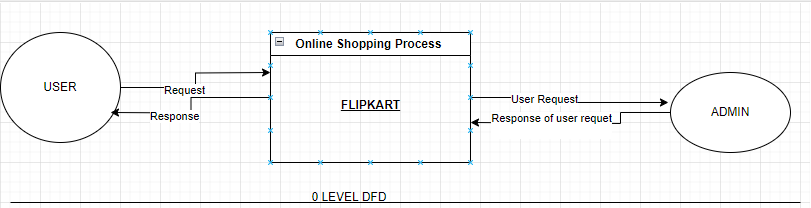
**6) Maintenance:**

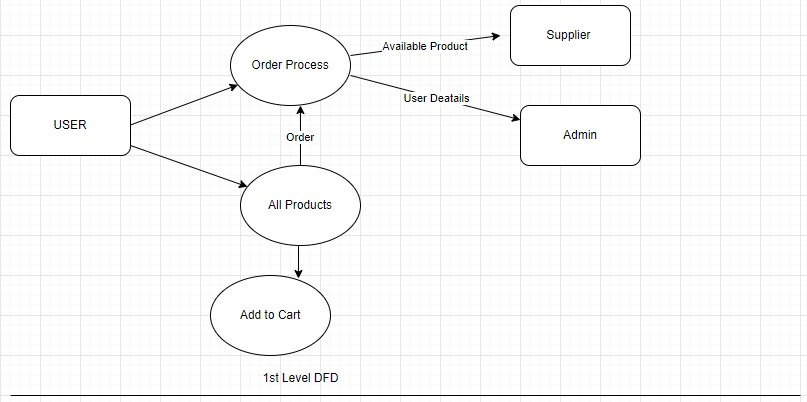
After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

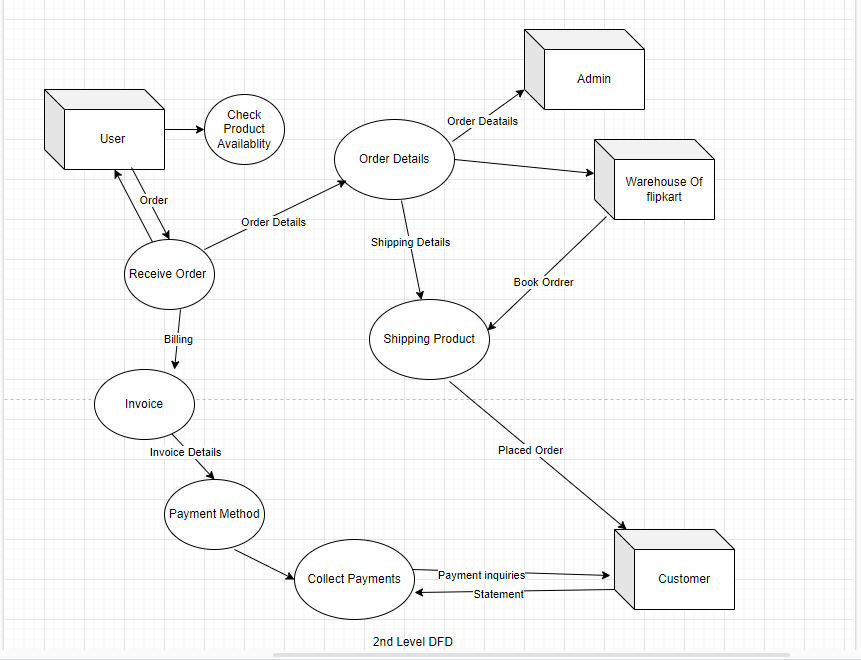
1. **What is DFD? Create a DFD diagram of Flipkart.**

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually “say” things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That’s why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

**Create a DFD diagram of Flipkart:**



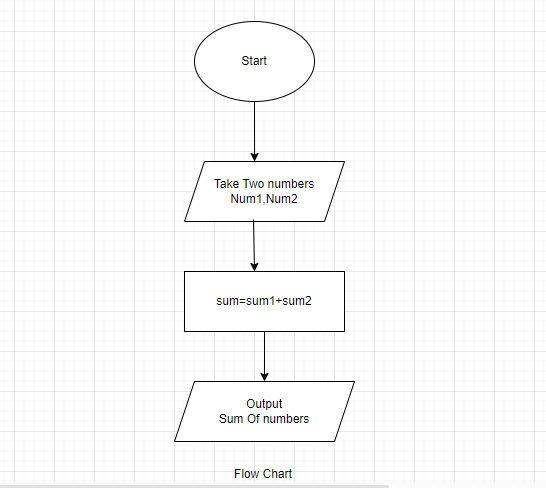




1. **What is Flow Chart? Create flow chart to make addition of two numbers.**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.

**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. It is a pictorial way of representing steps that are preferred by most beginner-level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. A flowchart is a picture of boxes that indicates the process flow sequentially. Since a flowchart is a pictorial representation of a process or algorithm, it’s easy to interpret and understand the process. To draw a flowchart, certain rules need to be followed which are followed by all professionals to draw a flowchart and are widely accepted all over the countries.



1. **What is Use case Diagram? Create use case on bill payment on Paytm.**

A Use Case Diagram is a vital tool in system design, it provides a visual representation of how users interact with a system. It serves as a blueprint for understanding the functional requirements of a system from a user’s perspective, aiding in the communication between stakeholders and guiding the development process.



USE CASE DIAGRAM BILL PAYMENT ON PAYTM.