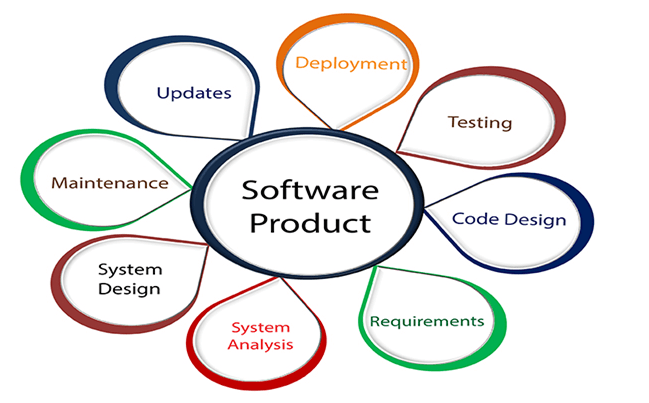
**Assignment :- 1**

**Q1. What is software? What is software engineering ?**

* What Is Software ?
* "**Software is a set of programs (sequence of instructions) that allows the users to perform a well-defined function or some specified task.**"
* 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 ?
* The term **software engineering** is the product of two words, **software**, and **engineering**.
* The **software** is a collection of integrated programs.
* Software subsists of carefully-organized instructions and code written by developers on any of various particular computer languages.
* Computer programs and related documentation such as requirements, design models and user manuals.
* **Engineering** is the application of **scientific** and **practical** knowledge to **invent, design, build, maintain**, and **improve frameworks, processes, etc**.



* **Software Engineering** is an engineering branch related to the evolution of software product using well-defined scientific principles, techniques, and procedures. The result of software engineering is an effective and reliable software product.
* **The importance of Software engineering is as follows:**
* **Reduces complexity:**
* **To minimize software cost:**
* **To decrease time:**
* **Handling big projects:**
* **Reliable software:**

**Q2. Explain types of software ?**

* Software's are broadly classified into two types, i.e., **System Software and Application Software**.
* System Software :-

1. Operating System

* Microsoft Windows
* Apple's iOS
* Apple's MacOS
* Android
* CentOS
* Linus
* Ubuntu
* Unix

### Device Drivers

* BIOS (Basic Input/Output System) Device Driver, Display Drivers
* Motherboard Drivers, USB (Universal Serial Bus) Drivers
* Printer Drivers, Sound Card Driver

### Firmware

* In electronic systems and computing, **firmware is a type of permanent software embedded in the system's ROM (read-only memory) to provide low-level control for some particular system device hardware**. It is a set of instructions that are stored permanently on your computer's hardware device.

Common examples of devices utilizing firmware are given below:

* Computer Peripherals
* Consumer Appliances
* Embedded Systems
* UEFI (United Extensible Firmware Interface)
* BIOS (Basic Input/Output System)

1. Utility

* **Utility software is developed to provide support in analyzing, optimizing, along configuring and maintaining a computer**. The job of the utility program is to offer support to the system infrastructure. Though the system will work even if it doesn't have any utility software, the right kind of utility software enhances its performance and makes it more reliable.

Some of the common examples of utility software are as follows:

* Norton and McAfee Antivirus
* WinRAR
* Directory Opus
* Disk defragmenter
* WinZip
* Windows File Explorer
* Razer Cortex
* Application Software :-
* **Application programs or software applications are end-user computer programs developed primarily to provide specific functionality to the user.** The applications programs assist the user in accomplishing numerous tasks such as doing **online research, completing notes, designing graphics, managing the finances, watching a movie, writing documents, playing games, and many more**. Therefore, many software applications are designed and developed every year by companies as per the demand and requirements of the potential users. The application software can either be designed for a general-purpose or specially coded as per the requirements of business cooperation.
* Today there are varieties of application software available in the market. Given below are some of the popular examples:

1. Word Processors

Word processor applications are globally **used for documentation, making notes, and typing data.** It also helps the end-users store and format data. They also enable the users to print their documents.

Some examples of Word Processor software's are as follows:

* MS Word (Microsoft)
* iWork-Pages (Apple)
* Corel WordPerfect
* Google Docs

2. Database Software

Database software is **used to create, manage, modify and organize a massive amount of data quickly retrieved.** Another name for database software **is Database Management System (DBMS).** Such software helps companies in their data organization. Common examples of Database Software's are:

* Oracle
* MS Access
* SQLite
* Microsoft SQL Server
* FileMaker
* dBase
* MariaDB
* MySQL

3. Multimedia Software

This software **enables the users to play, create or record images, music, and video files.** Different graphic designing companies widely use multimedia software to make animation, images, posts, packaging, marketing creative, gif, or even video editing. Due to their popularity and increasing demand, every software product development corporation has massive avenues in creating and upgrading them.

Common examples of Database Software's are given below:

* Adobe Photoshop
* Windows Movie Maker
* Adobe Illustrator
* Picasa
* Windows Media Player
* Corel Draw

4. Web Browsers

These are a type of software that is globally used to browse the Internet. **Web browsers help the users in positioning as well as fetching data across the web.** Common examples of web browsers are given below:

* Chrome
* Mozilla Firefox
* Microsoft Internet Explorer
* Opera
* Microsoft Edge
* UC Browser
* Apple Safari
* However, there also occurs **another classification of the software that exists on the basis of their availability and shareability.** The classification is given below:

### 1.Freeware

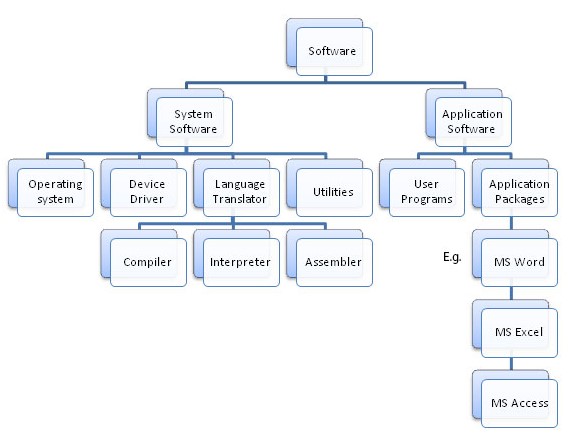
* Adobe Reader
* Zoom
* Skype
* ImgBurn
* Audacity
* Whatsapp
* Anydesk

### 2.Shareware

* Adobe Acrobat
* Adobe Photoshop
* AnyDVD
* PHP Debugger
* WinZip

### 3. Open-source

* Mozilla Firefox
* MySQL
* Thunderbird
* OpenOffice
* ClamWinantivirus
* Apache Web Server



**Q3. What is SDLC? Explain each phase of SDLC.**

* SDLC stands for Software Development Life Cycle, and it's a structured process for planning, writing, modifying, and maintaining software. The goal of SDLC is to produce high-quality, low-cost software that meets customer expectations in the shortest possible time.
* A software life cycle model (also termed process model) is a pictorial and diagrammatic representation of the software life cycle. A life cycle model represents all the methods required to make a software product transit through its life cycle stages. It also captures the structure in which these methods are to be undertaken.
* In other words, a life cycle model maps the various activities performed on a software product from its inception to retirement. Different life cycle models may plan the necessary development activities to phases in different ways. Thus, no element which life cycle model is followed, the essential activities are contained in all life cycle models though the action may be carried out in distinct orders in different life cycle models. During any life cycle stage, more than one activity may also be carried out.

### The 7 Phases Of SDLC (Software Development Life Cycle)

### 

#### ****Stage 1: Project Planning****

The first stage of SDLC is all about “What do we want?” Project planning is a vital role in the software delivery lifecycle since this is the part where the team estimates the cost and defines the requirements of the new software.

#### ****Stage 2: Gathering Requirements & Analysis****

The second step of SDLC is gathering maximum information from the client requirements for the product. Discuss each detail and specification of the product with the customer. The development team will then analyze the requirements keeping the design and code of the software in mind. Further, investigating the validity and possibility of incorporating these requirements into the software system. The main goal of this stage is that everyone understands even the minute detail of the requirement. Hardware, operating systems, programming, and [security](https://www.betsol.com/security-services/) are to name the few requirements.

#### ****Stage 3: Design****

In the design phase (3rd step of SDLC), the program developer scrutinizes whether the prepared software suffices all the requirements of the end-user. Additionally, if the project is feasible for the customer technologically, practically, and financially. Once the developer decides on the best design approach, he then selects the program languages like Oracle, [Java](https://www.betsol.com/blog/java-memory-management-for-java-virtual-machine-jvm/), etc., that will suit the software.

Once the design specification is prepared, all the stakeholders will review this plan and provide their feedback and suggestions. It is absolutely mandatory to collect and incorporate stakeholder’s input in the document, as a small mistake can lead to cost overrun.

#### ****Stage 4: Coding or Implementation****

Time to code! It means translating the design to a computer-legible language. In this fourth stage of SDLC, the tasks are divided into modules or units and assigned to various developers. The developers will then start building the entire system by writing code using the programming languages they chose. This stage is considered to be one of the longest in SDLC. The developers need certain predefined coding guidelines, and programming tools like interpreters, compilers, debugger to implement the code.

The developers can show the work done to the business analysts in case if any modifications or enhancements required.

#### ****Stage 5: Testing****

Once the developers build the software, then it is deployed in the testing environment. Then the testing team tests the functionality of the entire system. In this fifth phase of SDLC, the testing is done to ensure that the entire application works according to the customer requirements.

After testing, the [QA and testing](https://www.betsol.com/software-development-and-testing/) team might find some bugs or defects and communicate the same with the developers. The development team then fixes the bugs and send it to QA for a re-test. This process goes on until the software is stable, bug-free and working according to the business requirements of that system.

#### ****Stage 6: Deployment****

The sixth phase of SDLC: Once the testing is done, and the product is [ready for deployment](https://www.betsol.com/blog/how-to-make-software-deployments-easier/), it is released for customers to use. The size of the project determines the complexity of the deployment. The users are then provided with the training or documentation that will help them to operate the software.  Again, a small round of testing is performed on production to ensure environmental issues or any impact of the new release.

#### ****Stage 7: Maintenance****

The actual problem starts when the customer actually starts using the developed system and those needs to be solved from time to time. Maintenance is the seventh phase of SDLC where the developed product is taken care of. According to the changing user end environment or technology, the software is updated timely.

**Q4.What is DFD? Create a DFD diagram on Flipkart.**

A data flow diagram (DFD) represents the flow of information through a system. DFDs quickly became a popular way to visualize the major steps and data involved in software-system processes. They were used to show data flow in a software system, although they could, in theory, be applied to business process modeling. DFDs were useful to document the major data flows or to explore a new high-level design in terms of data flow.

## What is DFD?

A data-flow diagram (DFD) is a way of representing a flow of data of a process or a system (usually an information system), for example:

* Where Data Co­­mes From
* Where It Goes
* How It Gets Stored

In other words, it shows how data is processed by a system in terms of inputs and outputs. DFD is built using standardized symbols. There are several notations for displaying data-flow diagrams.

## History of DFD

The DFD notation draws on graph theory, originally used in operational research to model workflow in organizations. DFD originated from the Activity Diagram used in the SADT (Structured Analysis and Design Technique) methodology at the end of the 1970s. DFD popularizers include Edward Yourdon, Larry Constantine, Tom DeMarco, Chris Gane, and Trish Sarson.

### The Elements of a DFD

Different people use different notations to represent processes, data stores, data flow and external entities of the DFD. The most commonly used two different types of notations are by Yourdon & Coad or Gane & Sarson.

DFD uses 4 basic symbols to represent the flow of the diagram. They are

1. Process
2. Data Store
3. External Entity
4. Data Flow (Arrow)

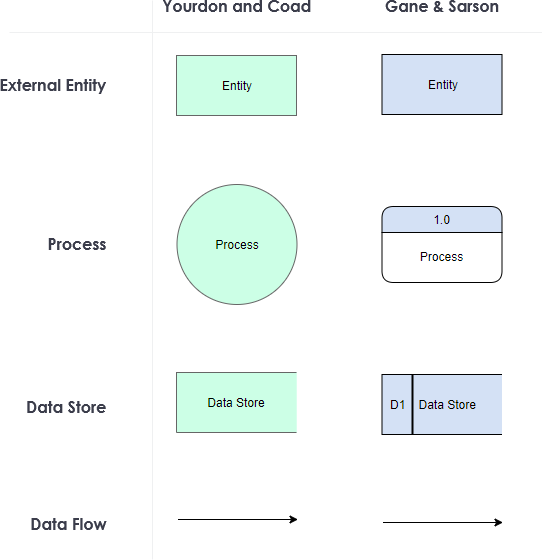
### Chris Gane & Trish Sarson Notation

The Gane &Sarson method stresses the identification of each node to help reference process narratives that further define each component either in plain text or pseudocode.

* A Data Store Is Represented As An Open Rectangle (3 Sides, Open On The Right).
* Processes Are Represented By A Rounded Upright Rectangle, Subdivided Into Sections For A Reference Value, A Process Description, And An Optional Location Specifier.

### Yourdon and Coad / DeMarco Notation

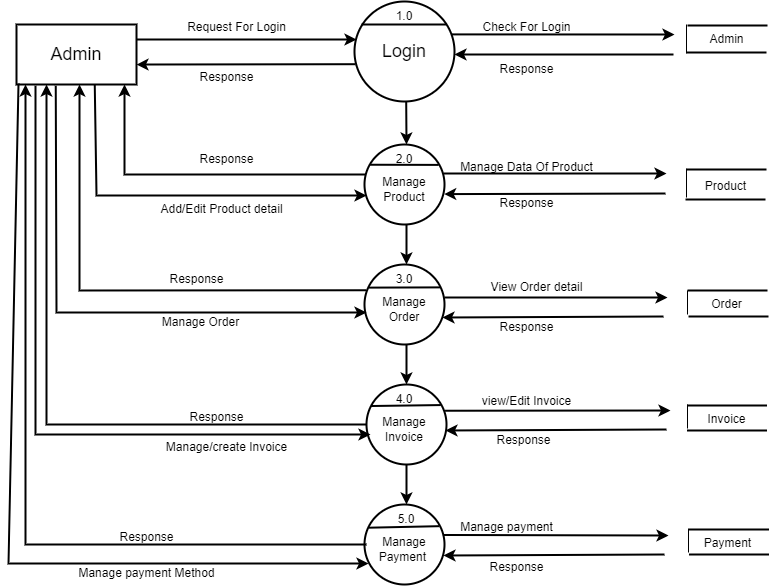
* Yourdon and Coad method includes components for creating data-flow diagrams and also object state diagrams. Some unique attributes of their DFD notation include:1) A Data Store Is Represented As Two Parallel, Horizontal Lines.2) Processes Are Represented By A Circle.



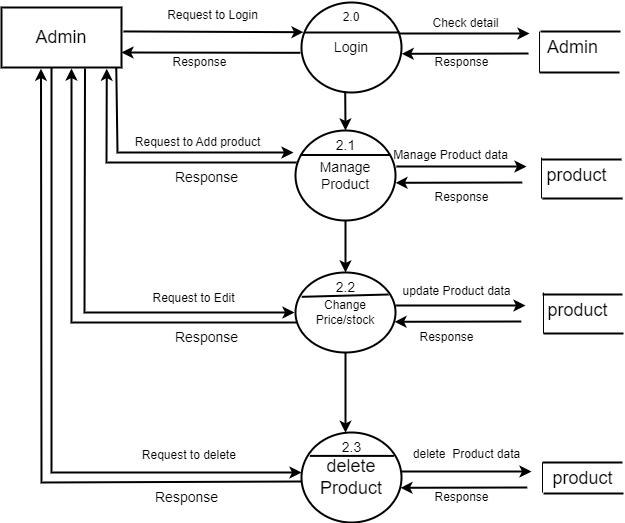
* **DFD diagram on Flipkart.**
* **0 lavel**

****

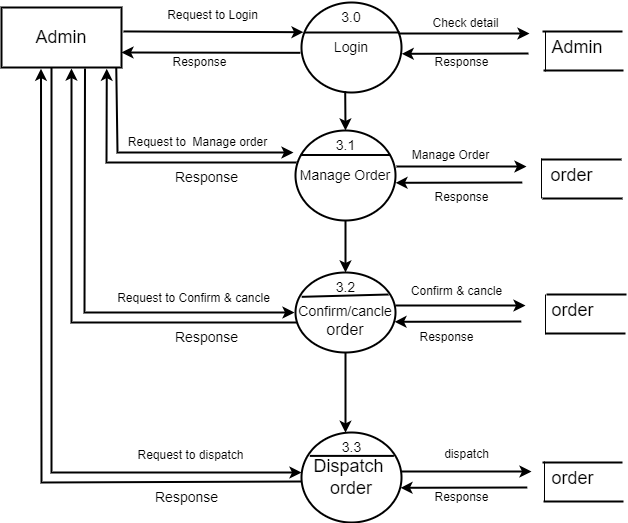
* **1-lavel**

****

* **2-lavel(1)**

****

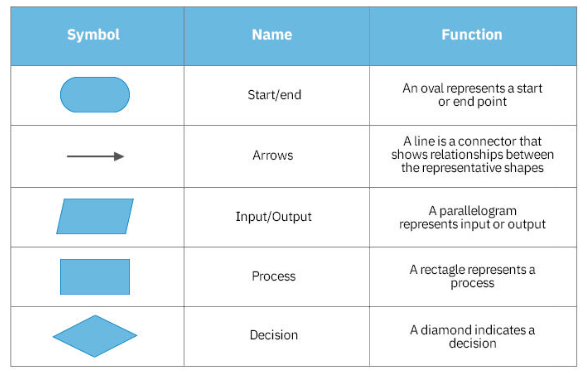
* **2-lavel(2)**

****

**Q5. What is Flow chart? Create a flowchart 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.

=algorithm=”  **A set of finite rules or instructions to be followed in calculations or other** problem-solving operations ”



#include<stdio.h>

Int main()

{

Int n1,n2,ans;

Printf(“Enter 2 values”);

Scanf(“%d%d”,&n1,&n2);

Ans=n1+n2;

Printf(“%d”,ans);

Return 0;

}

Algorithm:

1)start

2)declare variable n1,n2,ans;

3)display enter 2 values

4)read variable n1 and n2 from user

5)ans->n1+n2;

6)display ans

7)stop

### Flow Chart

### flowchart.png

**Q6. What is Use case Diagram? Create a use-case on bill payment on paytm.**

* UML Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

* Purpose of Use Case Diagrams

The main purpose of a use case diagram is to portray the dynamic aspect of a system. It accumulates the system's requirement, which includes both internal as well as external influences. It invokes persons, use cases, and several things that invoke the actors and elements accountable for the implementation of use case diagrams. It represents how an entity from the external environment can interact with a part of the system.

Following are the purposes of a use case diagram given below:

1. It gathers the system's needs.
2. It depicts the external view of the system.
3. It recognizes the internal as well as external factors that influence the system.
4. It represents the interaction between the actors.

