**Software Engineering Assignment**

**MODULE: 1 (SDLC)**

**Q.1 What is software? What is software engineering?**

* **Software** is more than just a program code. A program is an executable code, which serves some computational purpose. Software is considered to be collection of executable programming code, associated libraries and documentations. Software, when made for a specific requirement is called software product.
* **Software engineering** is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures. The outcome of software engineering is an efficient and reliable software product.

**Q.2 Explain types of software.**

There are two major software types in computer systems – System Software and Application Software.

1. **System Software**

System software allows the user to run computer software or hardware and is responsible for managing their interaction with each other. It perpetually runs in the background to maintain the basic functionalities and the hardware of the computer. It essentially acts as a mediator between the computer and the user to facilitate operations. System software is of four types, which are as follows-

* Operating System – The operating system is a collection of software that helps execute programs and offers a computer application’s general services. There are various types of operating systems, such as iOS, macOS, Windows, Unix, [Linux](https://www.simplilearn.com/linux-programming-for-beginners-article), and Ubuntu, among others.
* Device Drivers – Device drivers handle the operations of the hardware devices connected to a computer. They act as a software interface for the hardware devices so that applications and the operating system of a computer can run the hardware functions without having the knowledge of the hardware’s exact specifications.
* Firmware – Firmware is embedded in the read-only memory of a system and is a type of permanent software that offers low-level control for certain device hardware.
* Utility – Utility software functions to configure, maintain, and offer support in the analysis and optimization of the computer.

1. **Application Software**

[Application software](https://www.simplilearn.com/tutorials/programming-tutorial/what-is-application-software) or application programs are end-user programs that serve specific functionality to help users accomplish certain tasks. This includes graphic designing, researching online, drafting documents, watching movies or playing games, and managing finance among others. Developers keep creating software applications based on the evolving needs of users. There are various types of application software, which are as follows-

* Word Processors – Word processor software, as the name suggests, is designed for making notes, typing data, and documentation. Users can also format, store, and print their data and documents respectively using word processes.
* Database Software – Also known as [Database Management System](https://www.simplilearn.com/dbms-interview-questions-and-answers-article) (DBMS), database software allows users to perform various operations on quickly retrieved data, such as creation, management, organization, and modification. Some popular examples of DBMS are MS Access, dBase, [MySQL](https://www.simplilearn.com/tutorials/mysql-tutorial), Oracle, [Microsoft SQL Server,](https://www.simplilearn.com/what-is-microsoft-sql-server-architecture-article) and FileMaker.
* Multimedia Software – Users can perform certain actions on their computer, such as playing and recording music and video files as well as creating images with the help of multimedia software. This type of software has a massive application in the field of graphic designing where users create gifs, images, animations, and edit videos. Adobe Photoshop and Illustrator, Windows Movie Maker and Media Player, Corel Draw, and Picasa are some common types of multimedia processors.
* Web Browsers – Web Browsers are used for browsing the internet and their primary function is to allow users to fetch data across the web and also position the data. In simpler terms, you would access the internet with the help of web browsers so that you find the information that you need. Chrome, Opera, Mozilla Firefox, Microsoft Edge, Apple Safari, and UC Browser are very commonly used web browsers.
* Freeware – Freeware software is made available free of cost for a lifetime and it can be downloaded from the internet. This type of software is generally created by companies to improve their reach and gain more popularity. Some such software that is available free of cost for different purposes are Skype, Audacity, Zoom, Adobe Reader, WhatsApp, etc.
* Shareware – Shareware, much like Freeware, can be downloaded from the internet. However, the main difference between both is that, while freeware can be used for an unlimited amount of time without having to make any payment, shareware can be used on a trial basis. Adobe Acrobat and Photoshop, WinZip, and PHP Debugger are some popular types of shareware software.
* Open-Source – Open-source software is also available on the internet free of cost. However, what differentiates them from freeware is that they are available with their source code. This means users who download open-source software can make changes and transformations to it and even add features to it.

1. **Programming Software**

Programs and software are created by coders using different software tools, known as programming software. Some such programs used for software development by coders are as given below-

* Compilers – The conversion of codes written by humans into lower-level machine code is performed by compilers. These machine codes can be interpreted directly by computer hardware. While compilers serve a very basic purpose, they are the basis for creating even the most complicated and sophisticated software.
* Debuggers – Debuggers play an essential role in ensuring your software or application performs well by testing and debugging the computer code.
* Linkers – Linkers are responsible for combining various individual files from a compiler into a single executable file. The file converted, as a result, runs on its own without requiring a [programming](https://www.simplilearn.com/how-to-learn-programming-article) environment.
* Malware – Malware is software developed to attack computers and their software in a harmful way to cause them to misbehave or seize to work. This includes viruses, ransomware, trojans, and worms. Since there are a variety of malware that may be mistakenly downloaded, it is crucial to have antimalware software on your computer to keep it safe from their attacks.

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

Software Development Life Cycle, SDLC for short, is a well-defined, structured sequence of stages in software engineering to develop the intended software product.   
  
SDLC Activities SDLC provides a series of steps to be followed to design and develop a software product efficiently. SDLC framework includes the following steps:

**Communication**

This is the first step where the user initiates the request for a desired software product. The user contacts the service provider and tries to negotiate the terms, submits the request to the service providing organization in writing.

**Requirement Gathering**

This step onwards the software development team works to carry on the project. The team holds discussions with various stakeholders from problem domain and tries to bring out as much information as possible on their requirements. The requirements are contemplated and segregated into user requirements, system requirements and functional requirements. The requirements are collected using a number of practices as given –

* studying the existing or obsolete system and software,
* conducting interviews of users and developers,
* referring to the database or
* collecting answers from the questionnaires.

**Feasibility Study**

After requirement gathering, the team comes up with a rough plan of software process. At this step the team analyzes if a software can be designed to fulfill all requirements of the user, and if there is any possibility of software being no more useful. It is also analyzed if the project is financially, practically, and technologically feasible for the organization to take up. There are many algorithms available, which help the developers to conclude the feasibility of a software project.

**System Analysis**

At this step the developers decide a roadmap of their plan and try to bring up the best software model suitable for the project. System analysis includes understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc. The project team analyzes the scope of the project and plans the schedule and resources accordingly.

**Software Design**

Next step is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered in requirement gathering phase are the inputs of this step. The output of this step comes in the form of two designs; logical design, and physical design. Engineers produce meta-data and data dictionaries, logical diagrams, data-flow diagrams, and in some cases pseudo codes.

**Coding**

This step is also known as programming phase. The implementation of software design starts in terms of writing program code in the suitable programming language and developing error-free executable programs efficiently.

**Testing**

An estimate says that 50% of whole software development process should be tested. Errors may ruin the software from critical level to its own removal. Software testing is done while coding by the developers and thorough testing is conducted by testing experts at various levels of code such as module testing, program testing, product testing, in-house testing, and testing the product at user’s end. Early discovery of errors and their remedy is the key to reliable software.

**Integration**

Software may need to be integrated with the libraries, databases, and other program(s). This stage of SDLC is involved in the integration of software with outer world entities.

**Implementation**

This means installing the software on user machines. At times, software needs post-installation configurations at user end. Software is tested for portability and adaptability and integration related issues are solved during implementation.  
  
**Operation and Maintenance**

This phase confirms the software operation in terms of more efficiency and less errors. If required, the users are trained on, or aided with the documentation on how to operate the software and how to keep the software operational. The software is maintained timely by updating the code according to the changes taking place in user end environment or technology. This phase may face challenges from hidden bugs and real-world unidentified problems.

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

Data Flow Diagram (DFD) is a graphical representation of flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow, and stored data. The DFD does not mention anything about how data flows through the system.

There is a prominent difference between DFD and Flowchart. The flowchart depicts flow of control in program modules. DFDs depict flow of data in the system at various levels. It does not contain any control or branch elements.

**Types of DFD**

Data Flow Diagrams are either Logical or Physical.

**Logical DFD** - This type of DFD concentrates on the system process, and flow of data in the system. For example, in a banking software system, how data is moved between different entities.

**Physical DFD** - This type of DFD shows how the data flow is actually implemented in the system. It is more specific and closer to the implementation.

**DFD Components**

DFD can represent source, destination, storage, and flow of data using the following set of components –

**Entities** - Entities are sources and destinations of information data. Entities are represented by rectangles with their respective names.

**Process** - Activities and action taken on the data are represented by Circle or Round-edged rectangles.

**Data Storage** - There are two variants of data storage - it can either be represented as a rectangle with absence of both smaller sides or as an open-sided rectangle with only one side missing.

**Data Flow** - Movement of data is shown by pointed arrows. Data movement is shown from the base of arrow as its source towards head of the arrow as destination.

Levels of DFD

**Level 0** - Highest abstraction level DFD is known as Level 0 DFD, which depicts the entire information system as one diagram concealing all the underlying details. Level 0 DFDs are also known as context level DFDs.

**Level 1** - The Level 0 DFD is broken down into more specific, Level 1 DFD. Level 1 DFD depicts basic modules in the system and flow of data among various modules. Level 1 DFD also mentions basic processes and sources of information.

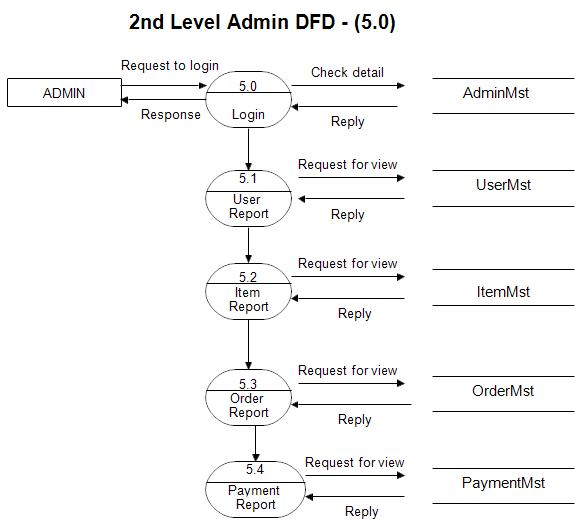
**Level 2** - At this level, DFD shows how data flows inside the modules mentioned in Level 1.

Higher level DFDs can be transformed into more specific lower level DFDs with deeper level of understanding unless the desired level of specification is achieved.

Data Flow Diagram of Flipkart is as per below images from the Admin side and User side:

  
  
**A picture containing text, screenshot, receipt, diagram

Description automatically generated  
A picture containing text, screenshot, font, receipt

Description automatically generatedA picture containing text, diagram, receipt, parallel

Description automatically generatedA picture containing text, receipt, diagram, font

Description automatically generatedA picture containing text, diagram, receipt, screenshot

Description automatically generated**

**Q.4 What is Flowchart? Create a flowchart to make addition of two numbers.**

A flowchart is a graphical representation of a process or algorithm that uses various symbols and arrows to depict the sequence of steps. It is commonly used in computer programming, problem-solving, and process documentation. A flowchart helps in understanding the logical flow of a task or program.

Here is a simple flowchart to demonstrate the addition of two numbers:

**```**

**Start**

**|**

**v**

**Input the first number (A)**

**|**

**v**

**Input the second number (B)**

**|**

**v**

**Add A and B**

**|**

**v**

**Display the result**

**|**

**v**

**Stop**

**```**

In this flowchart, the process begins with the "Start" symbol and then proceeds to input the first number (A) and the second number (B). After that, the two numbers are added together, and the result is displayed. Finally, the flowchart reaches the "Stop" symbol, indicating the end of the process.

Note that the flowchart symbols used in this example are the standard symbols commonly used in flowchart diagrams. The "Input" and "Display" steps can be represented using parallelograms, and the "Start," "Stop," and the addition step can be represented using rectangles. The arrows depict the flow of control from one step to another.

**Q.5 What is Use Case Diagram? Create a use-case on bill payment on paytm.**

Use case diagrams are visual representations that illustrate the interactions between users (actors) and a system. They show the different scenarios (use cases) in which the system is involved, helping to understand the system's functionality from a user's perspective.

Actors represent users or external systems, while use cases represent specific functionalities or behaviors of the system. Relationships depict associations between actors and use cases.

Use case diagrams are useful for capturing requirements and communicating the system's behavior without delving into internal details. They are commonly used during the analysis and design phases of software development.

