**MODULE: 1**

**SE – Overview of IT Industry**

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

**Ans : definition of software :**

Software is a set of instructions, data or programs used to operate computers and execute specific tasks.

**Definition of software engineering :**

Software engineering is the branch of computer science that deals with the design, development, testing, and maintenance of software applications.

**2. Explain types of software**

**Ans:**

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](https://www.techtarget.com/searchcio/feature/The-rise-of-modern-applications-Why-you-need-them) 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](https://www.techtarget.com/whatis/definition/firmware), computer language translators and system [utilities](https://www.techtarget.com/whatis/definition/utility).
3. **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.
4. **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.
5. **Programming software.** Computer programmers use programming software to write code. Programming software and programming tools enable developers to develop, write, test and [debug](https://www.techtarget.com/searchsoftwarequality/definition/debugging) other software programs. Examples of programming software include assemblers, compilers, debuggers and interpreters.

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

**Ans:**

**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.

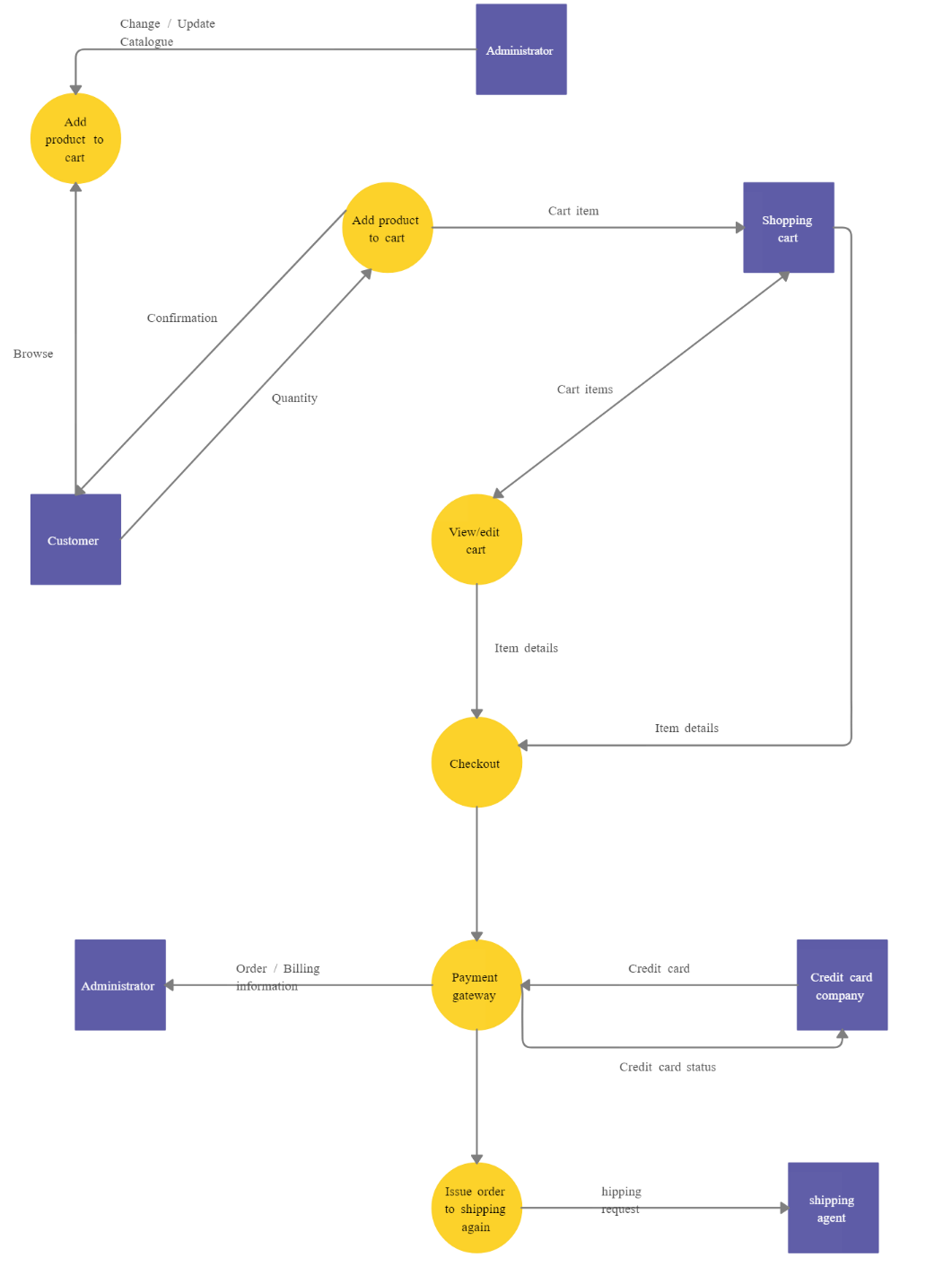
**6. Maintenance:**This phase includes ongoing support, bug fixes, and updates to the software.

There are **different**[**methodologies**](https://www.geeksforgeeks.org/5-most-commonly-used-software-development-methodologies/?ref=lbp) that organizations can use to implement the SDLC, such as**Waterfall, Agile, Scrum, V-Model**and**DevOps.**

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

**Ans:**

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement.

****

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

**Ans:**   
A flowchart is a visual representation of a process or algorithm, using standardized symbols to depict different steps, decisions, and the flow of data. It is a diagrammatic representation that helps illustrate the sequence of operations in a system, making complex processes more understandable. Flowcharts are widely used in various fields, including software development, business processes, engineering, and education.



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

A use case diagram is a type of Unified Modeling Language (UML) diagram used to visually represent the interactions between different actors (users or external systems) and a system. It provides a high-level view of the functionality and requirements of a system from the perspective of its users. Use case diagrams are commonly used in software development and system design to capture and communicate the behavioral aspects of a system.