**🎯 *Module–1(Fundamental)* 🎯**

**Que.1) What is Software Testing?**

Ans. Software testing is a process of executing a program or application with the intent of finding software bugs.

In other words,

* Software testing is a process used to identify the correctness, completeness, and quality of developed computer software.
* Software testing is a method to check whether *the* **actual software product matches** **expected as per client requirements** and to ensure that software product is defect free.

**Que.2) What is SDLC?**

Ans. SDLC stands for Software development life cycle.

**🎯** SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment and ongoing maintenance and support.

**🎯** There are a number of different development models.

**Que.3) Write SDLC Phases with basic requirements**

Ans. SDLC Phases below following are:

**🎯** Requirements Collection/Gathering

**🎯** Analysis

**🎯** Design

**🎯** Implementation

**🎯** Testing

**🎯** Maintenance

* **Requirement Collection/Gathering**: Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.

➢Three types of problems can arise:

➢ Lack of clarity: It is hard to write documents that are both precise and easy-to-read.

➢ Requirements confusion: Functional and Non-functional

➢ Requirements Amalgamation: Several different requirements

➢ Types of Requirements:

➢ Functional Requirements: describe system services or functions.

➢ Compute sales tax on a purchase Update the database on the server

➢ Non-functional requirements: are constraints on the system or the development process.

➢ Non-functional requirements may be more critical than functional requirements.

* **Analysis Phase**

➢This analysis represents the “what” phase.

➢The requirement documentaries to capture the requirements from the customer's perspective by defining goals.

➢The architecture defines the components, their interfaces and behaviours.

* **Design Phase**

➢Design Architecture Document Implementation Plan.

➢Critical Priority Analysis Performance Analysis Test Plan.

➢The architecture team also converts the typical scenarios into a test plan.

* **Implementation Phase**

➢In the implementation phase, the team builds the components either from scratch or by composition.

➢The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.

* **Testing Phase**

➢Quality is a distinguishing attribute of a system indicating the degree of excellence.

➢It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.

* **Maintenance Phase**

➢Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.

➢ Maintenance is the three types of process changing a system after it has been development.

➢ Corrective maintenance: identifying and repairing defects.

➢ Adaptive maintenance: adapting the existing solution to the new platforms.

➢ Perfective Maintenance: implementing the new requirements.

**Que.4) Explain Phases of the waterfall model**

Ans. There are main 2 types of Phases of waterfall model:

1. Verification Phase (Development Levels)
2. Validation Phase (Test Levels)
3. Verification Phase

**🎯** Business Requirement Analysis**:** This is the first phase in the development cycle where the product requirements are understood from the customer’s perspective.

**🎯** System Design (System Requirement): Once you have clear and detailed product requirements, it’s time to design the complete system.

**🎯** Architectural Design (Technical Specification): Architectural specifications are understood and designed in this phase.

**🎯** Module Design (Program Specification): In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD).

**🎯** Code Phase: The actual coding of the system modules designed in the design phase is taken up in the Coding phase.

1. Validation Phase

**🎯** Unit Testing: Unit tests designed in the module design phase are executed on the code during this validation phase.

➢Unit testing is the testing at the code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

**🎯** Integration Testing: Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.

**🎯** System Testing: System testing is directly associated with the System design phase. System tests check the entire system functionality and the communication of the system under development with external systems.

**🎯** Acceptance Testing: Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment.

**Que.5)** **Write phases of spiral model**



🎯There are **4 Phases** of Spiral Model following are:

1. Planning: Determination of objectives, alternatives and Constraints.
2. Risk Analysis: Analysis of alternatives and identification/ resolution of risks.
3. Customer Evaluation: Assessment of the results of engineering.
4. Engineering: Development of the “next level” product.

**Que.6) What is SRS?**

**Ans.** SRS stands for “Software Requirements Specification.”

➢A software requirements specification (SRS) is a complete description of the behaviour of the system to be developed.

➢It includes a set of use cases that describe all of the interactions that the users will have with the software.

➢This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

**Que.7) What is agile methodology?**

Ans. Agile is a simple and ingenious concept through continuous incremental through small and frequent releases.

➢Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.

➢In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

**Que.8) Write agile manifesto principles**

**Ans.** Highest priority is to satisfy the customers to early continue delivery software.

➢Being flexible about changing requirements at any point of development.

➢Face to face communication as the most effective way to communicate between customer and development team.

➢Transparency between business people and developers and requires them to work together.

**Que.9) Explain working methodology of agile model and also write pros and cons.**

**Ans.** Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

➢ Agile Methods break the product into small incremental builds. These builds are provided in iterations.

➢Each iteration typically lasts from about one to three weeks.

➢Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

**🎯Pros(Advantages):**

➢Agile model a very realistic approach to software development Promotes teamwork and cross training.

➢Suitable for fixed or changing requirements delivers early partial working solutions.

➢Enables concurrent development and delivery within an overall.

➢Planned context.

➢Little or no planning required Easy to manage.

➢ Gives flexibility to developers.

**🎯Cons(Disadvantages):**

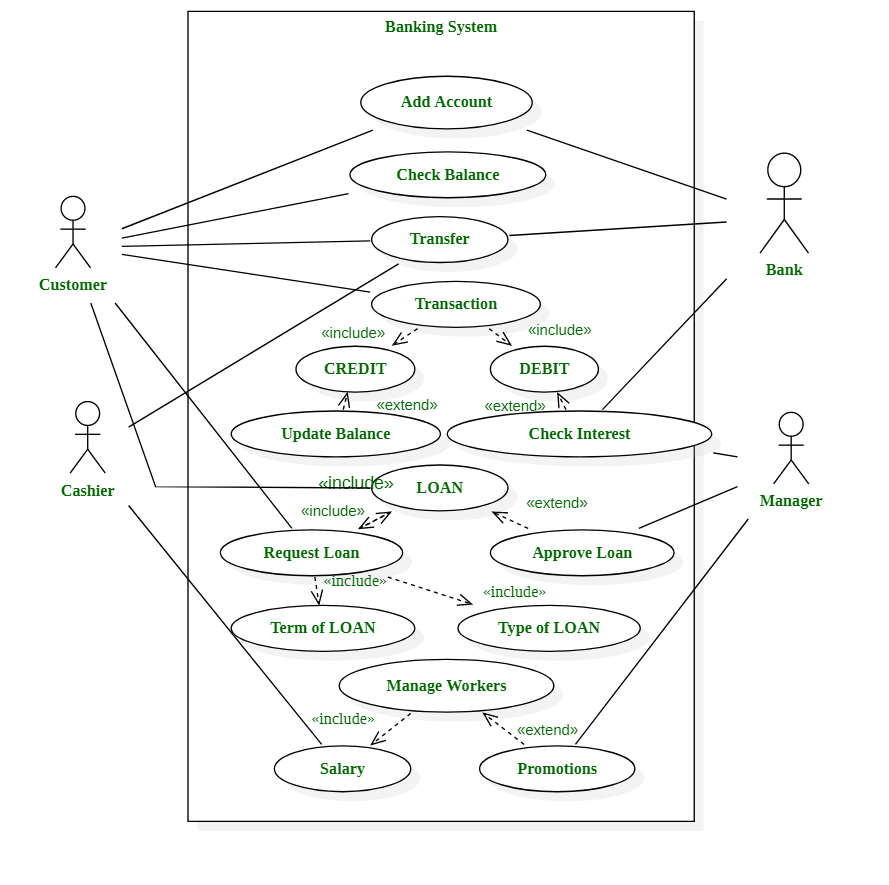
➢Not suitable for handling complex dependencies.

➢More risk of sustainability, maintainability and extensibility.

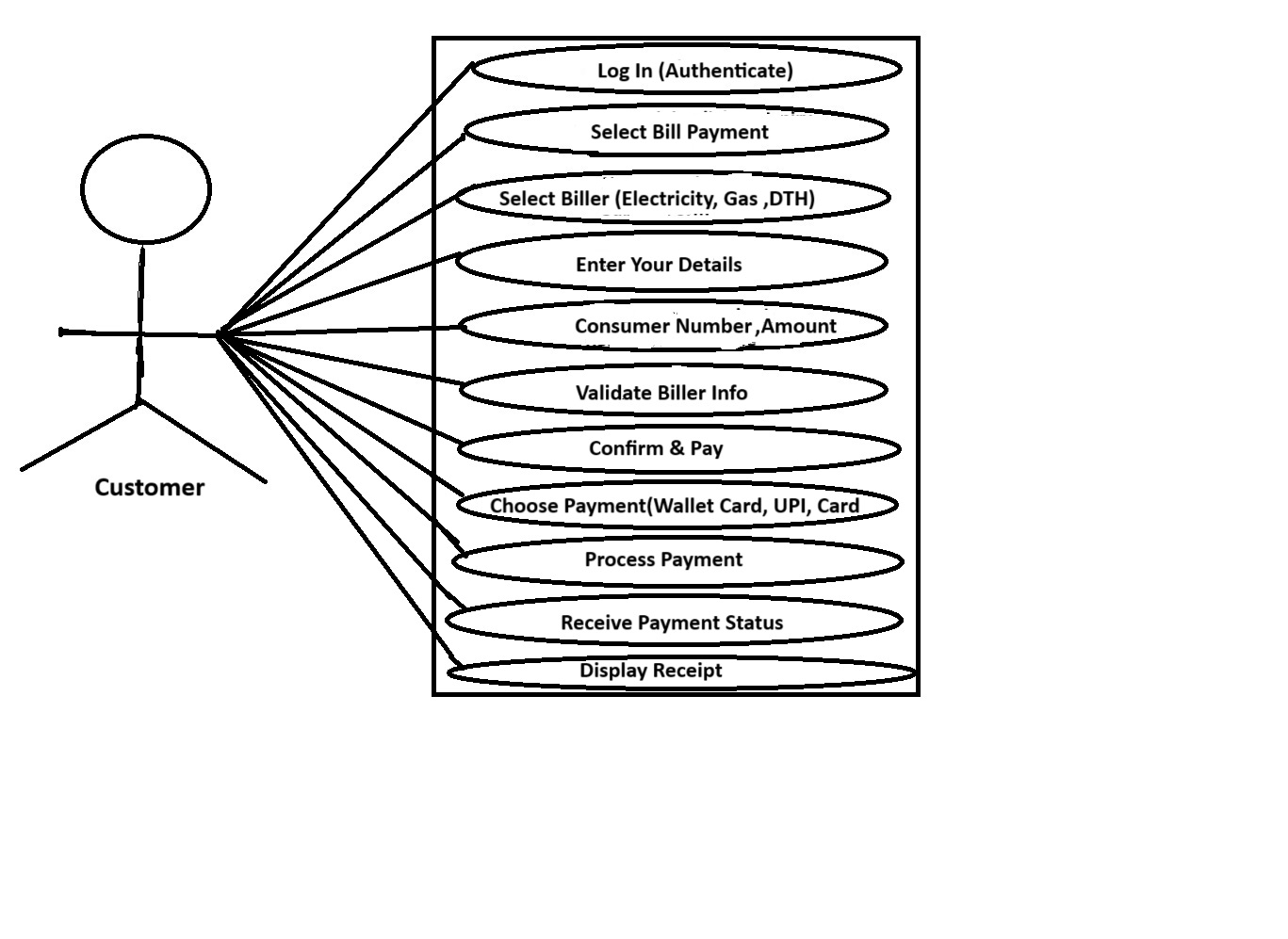
➢An overall plan, an agile leader and agile PM practice is a must without which it will not work.

➢Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

**Que.10) Draw Usecase on banking system for customers**.

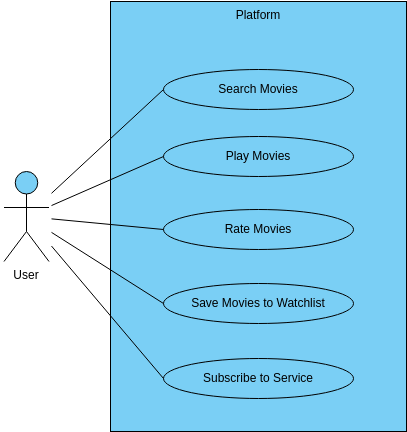
Ans. 

**Q.11) Draw Usecase on online bill payment system (paytm)**

Ans.

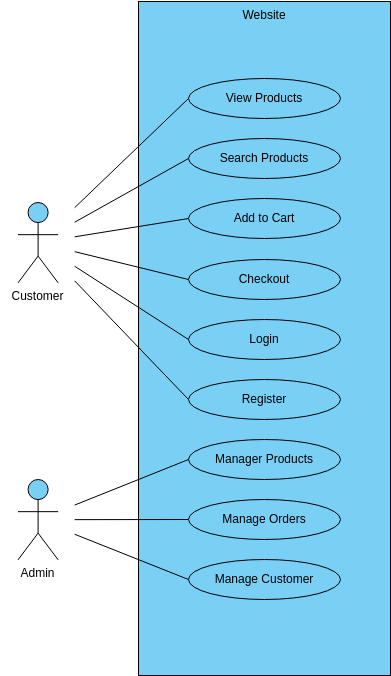
**Que.12)** **Draw Usecase on OTT plateform.**

**Ans.**

****

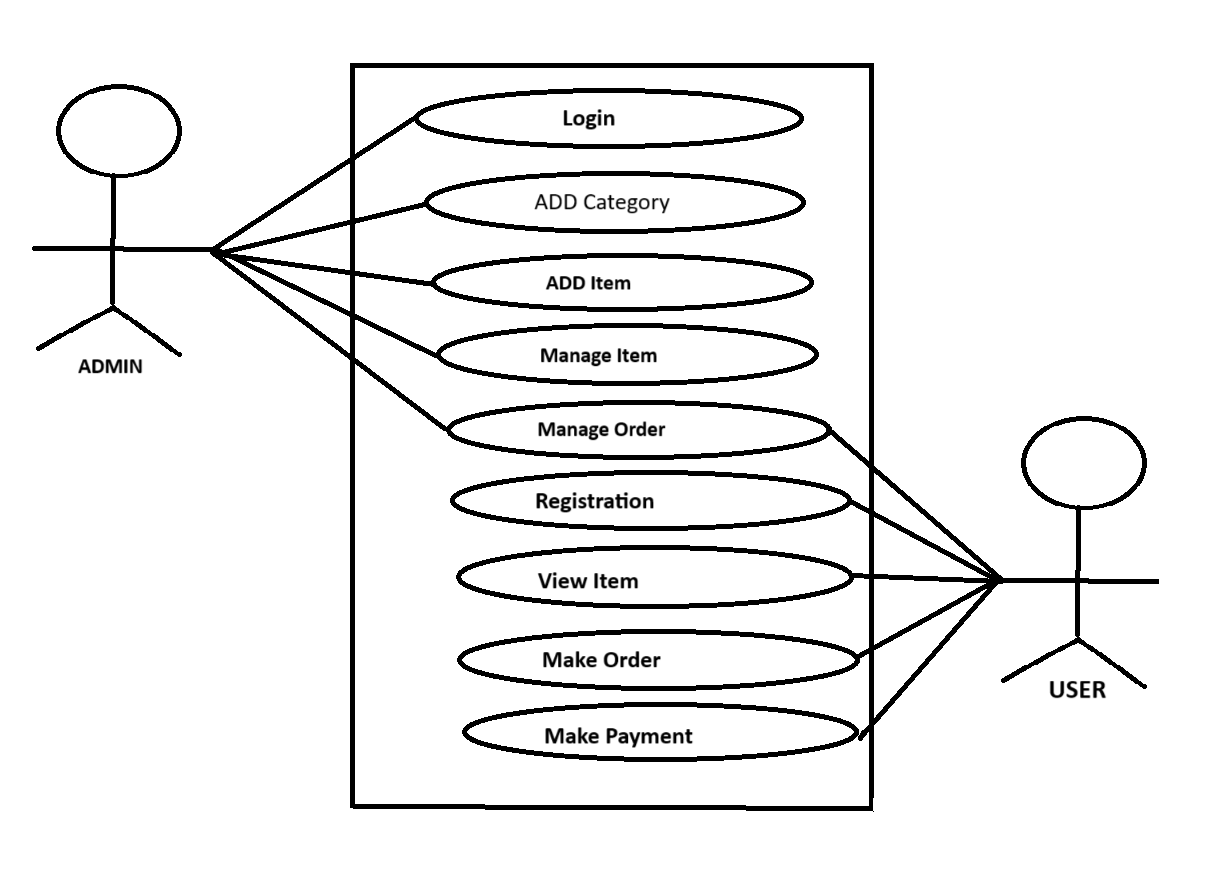
**Que.13) Draw usecase on E-commerce application**

Ans.



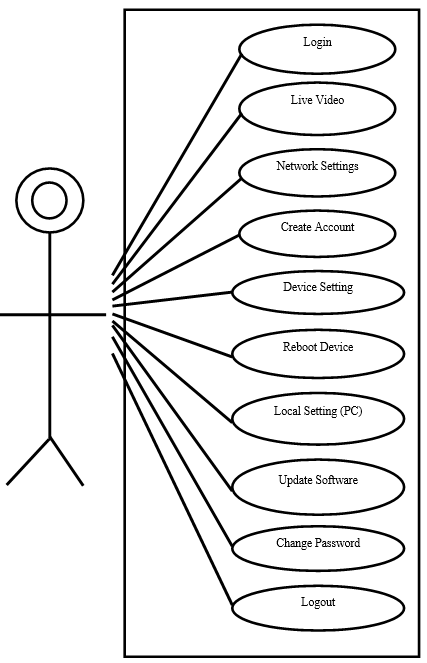
**Que.14) Draw usecase on Online shopping product using payment gateway.**

**Ans.**



**Que.15) Draw Use Case on Broadcasting System.**

**Ans.**

****

**Que.16) What is OOPS?**

**Ans.** OOPS is stands for “**O**bject **O**riented **P**rogramming **S**ystem.”

➢Programming is like writing.

➢If you can write a demonstration, you can make a program. So, programming is also easy.

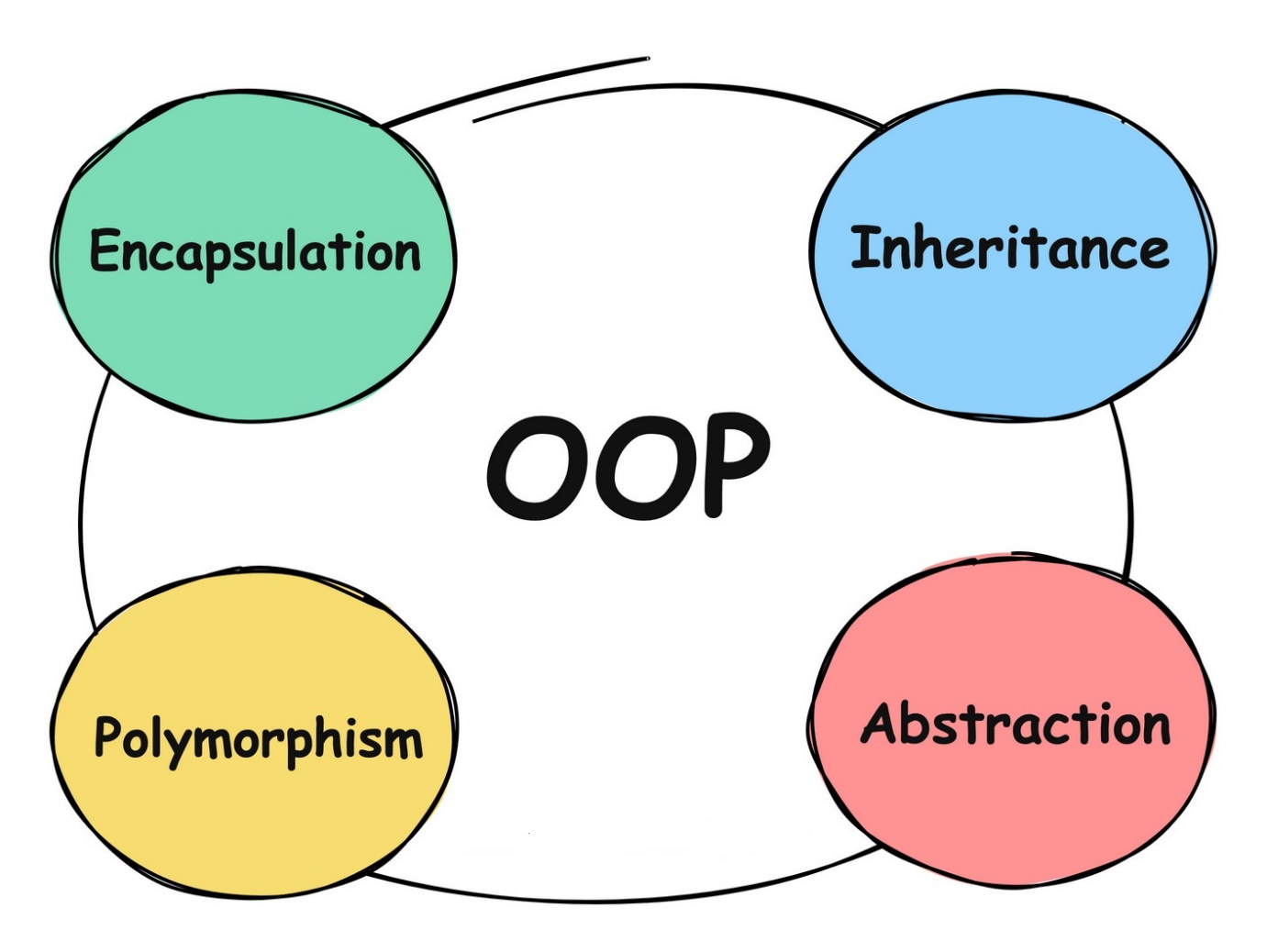
➢But, actually, programming is not so easy, because a real good program is not easily programmed. It needs the programmers’ lots of wisdom, lots of knowledge about programming and lots of experience.

➢It is like writing, to be a good writer needs lots of experience and lots of knowledge about the world. Learning and practise is necessary.

**Que.17) Basic concept of oops**

**Ans.** Object-Oriented Programming (OOP) is a programming paradigm built around **objects**—data structures combining **state** (attributes) and **behaviour** (methods).

➢The key to OOP lies in its **four core principles**, often called the “four pillars”:

****

➢ Four Pillars in brief following are:

**1. Abstraction:**

➢You **simplify complexity** by exposing only essential features and hiding internal details.

**🎯** Think of a car: you just use a steering wheel and pedals—you don’t need to understand the engine internals.

### 2. Encapsulation

➢You **bundle data and methods** into a single unit (class), **restricting access** to internal representation.

**🎯** Like a TV remote—you can press buttons, not tamper with internal circuits.

**🎯** Classes use access modifiers (private, public, etc.) to control what’s exposed.

### 3. 👥 Inheritance

You establish an **“is-a” relationship**, allowing a class to **reuse attributes and behaviours** from another.

**🎯** Example**: class Car → class Tesla extends Car;** Tesla has all Car features plus new ones.

**🎯**This promotes **code reuse** and reduces duplication.

### 4. *🌀* Polymorphism

You allow a single interface or method to **work in multiple ways** depending on context.

**🎯** E.g., a method draw() can behave differently if called on a Circle or a Square.

**🎯** Enables flexibility and dynamic behavior: “one name, many forms.”

**Que.18) What is Object?**

**Ans.** This is the basic unit of object oriented programming(OOP).

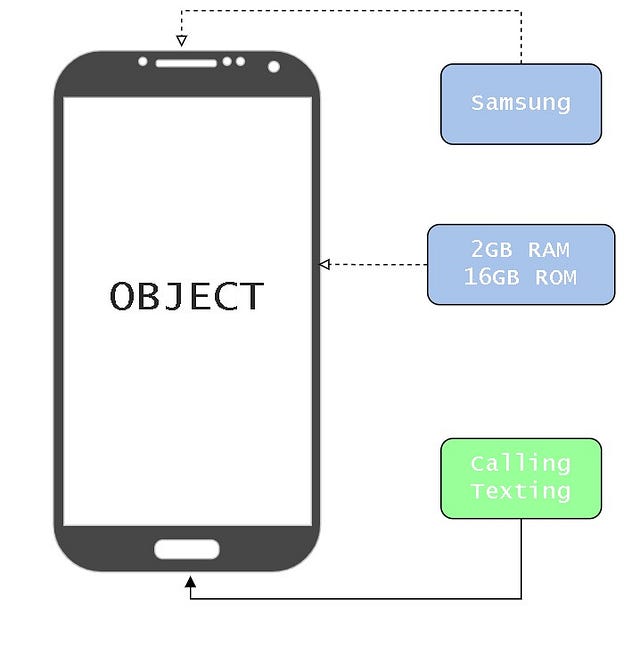
➢An Object is a single instance of a class, which contains data and methods on that data. So an object consists of three things:

**🎯** Name: This is a variable name that represents the object.

**🎯** Member data: The data that describes the object.

**🎯** Member method: Behavior that describes the Object.

**For Exam. – Samsung Galaxy** is an object with the brand name Samsung, 2 GB RAM as properties, and calling and texting as behavoir.



**Que.19) What is class?**

**Ans.** Class is an collection of data member(Variable) and member function(method or process) with its behavior.

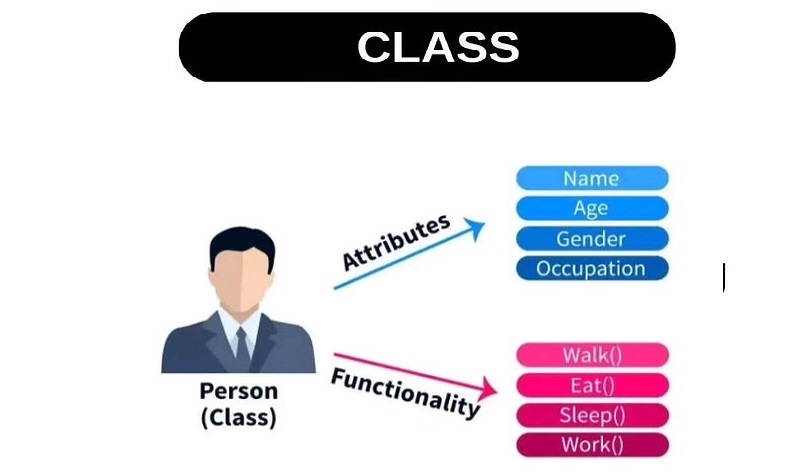
**Syntax :** Class classname

**{**

data members

Member Function

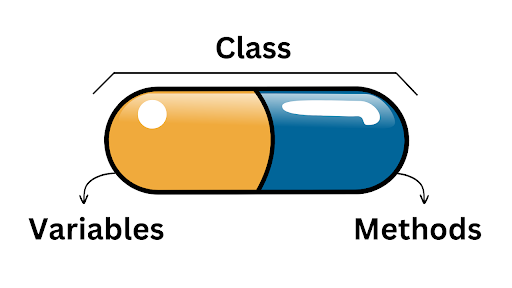
**}**



**Que.20) What is encapsulation?**

**Ans.** Encapsulation(data hiding) Wrapping up of data into single unit.

➢Private your data member and member function.

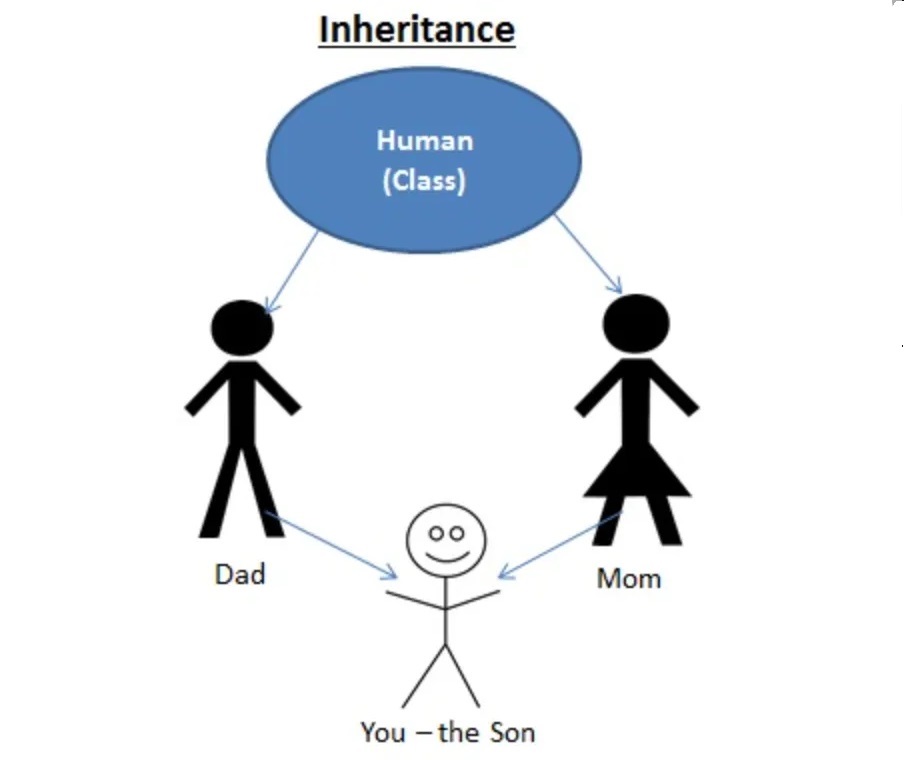


**Que.21) What is inheritance?**

**Ans. Inheritance :**Itis a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of [OOPs](https://www.javatpoint.com/java-oops-concepts) (Object Oriented programming system).

**🎯 Subclass** (child) — the class that inherits from another class. It’s also called a derived class, extended class, or child class.

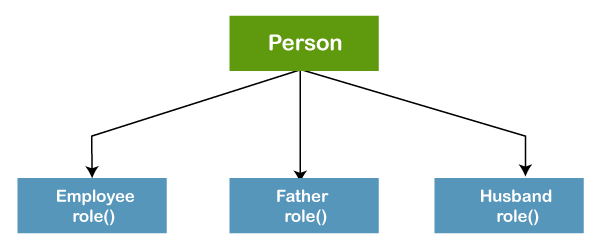
**🎯 Superclass** (parent) — the class being inherited from. It is also called a base class or a parent class.



**Que.22) What is polymorphism?**

**Ans.** Polymorphism means “having many forms”.

➢It allows different objects to respond to the same message in different ways, the response specific to the type of the object.

****

**s**