

MODULE-1 ASSIGNMENT OF SOFTWARE TESTING

1. What is SDLC?

- The software Development Life Cycle is a Structured Process that enables the Production of high-quality, low-cost Software, in the shortest possible production time.
- A Software Development Life Cycle is essentially a series of steps, or Phases, that provide a model for the development and lifecycle management of an application or piece of software.

2. What is Software Testing?

- Software Testing is a process used to identify the correctness, completeness and quality of developed computer Software.
- Software testing is the part of software development process.

3. What is agile methodology?

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and consumer satisfaction by rapid delivery of working software product.
- Agile methods break the product into small incremental builds. These builds are provided in iterations.
- 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 decided to time boxes (small time frames) to deliver specific features for a release.

4. What is SRS?

- SRS (Software Requirements Specification) is a complete description of the behavior of the system to be developed.
- SRS includes a set of cases that described all of the interactions that the users will have with the software.
- Use cases are also known as fundamental requirements. In addition to use cases, the SRS also contains non-functional (or supplementary) requirements.
- Non-functional requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

5. What is OOP?

- Identifying objects and assigning responsibilities to these objects.
- Objects communicate to other objects by sending messages.
- Messages are received by the methods of an object.
- An object is like a black box.
- The internal details hidden.
- Object is derived from abstract data type.
- Object oriented programming has a web of interacting objects, each house keeping its own state.
- Object of a program interact by sending messages to each other.

6. Write Basic Concepts of OOPS

- Object
- Class
- Encapsulation
- Inheritance
- Polymorphism
 - Overriding
 - Overloading
- Abstraction

7. What is Object?

- Any entity which has own state and behavior that is called an object. (ex. Any living thing)
- An object represent as individual, identifiable item, unit or entity either real or abstract, with a well-defined roll in the problem domine.
- An object is anything to which concepts applies.
- That is both Data and Function that operate on data are bundled as a unit called as object.

8. What is Class?

- Collection of objects (ex. Human body)
- When you defined a class, you defined a blueprint for an object.
- A class represents an obstruction of the object and abstracts the property and behavior of that object.
- Class can be considered as the blueprint or definition or a template for an object and describes the properties and behavior of that object, but without any actual existence.

9. What is Encapsulations?

- Binding of data – wrapping up of data. (ex. Capsule)
- Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.
- Encapsulation in java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit here is a class (or interface).
- Encapsulation in plain English means to enclose or be enclosed in or as if in a capsule. In java a class is a capsule (or unit).

10. What is Inheritance?

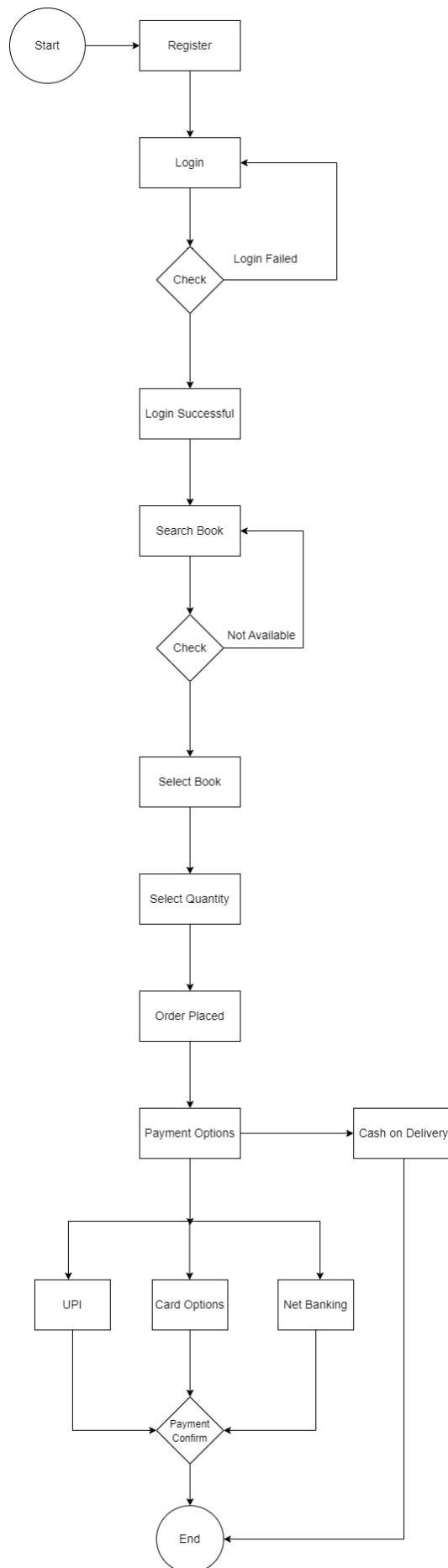
- Inheritance means that one class inherits, the characteristics of another class. This is also called a “is a” relationship.
- One of the most useful aspects of object-oriented programming is code reusability. As the name suggests inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.
- This is a very important concept of object-oriented programming since this feature helps to reduce the code size.
- Inheritance describes the relationship between two classes. A class can get some of its features from a parent class and then add unique features of its own.

11. What is Polymorphism?

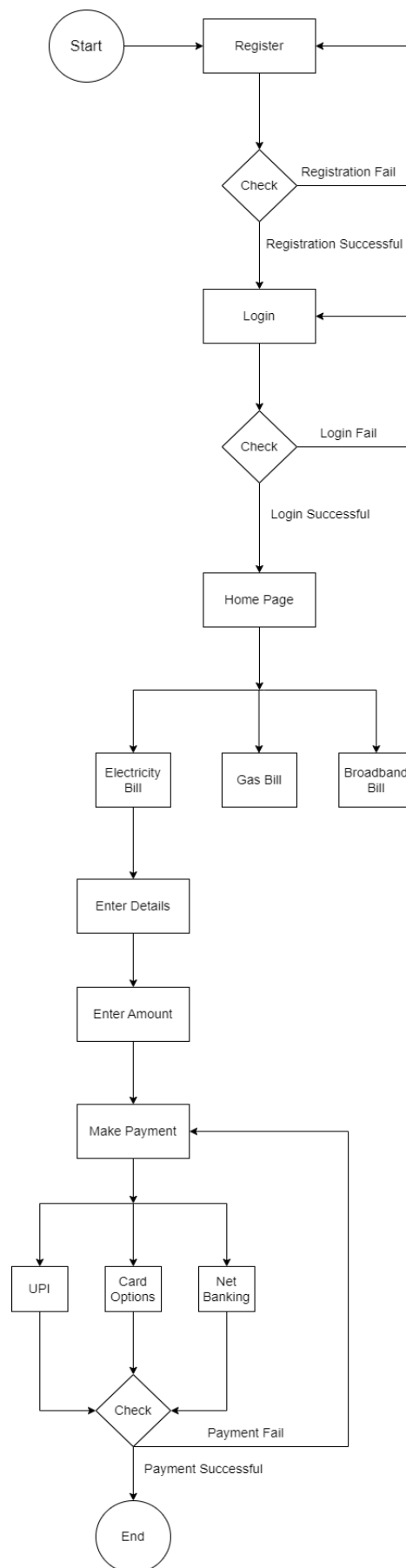
- Polymorphism means “having many forms”.

- It allows different objects to respond to the same message in different ways, the response specific to the type of the object.
- The most important aspect of an object is its behavior (the things it can do). A behavior is initiated by sending a message to the object (usually by calling a method).
- The ability to use an operator or function in different ways in other words giving different meaning or functions to the operations or functions is called polymorphism.

12. Draw use case on Online book shopping.



13. Draw use case on online bill payment system (paytm)



14. Write SDLC phases with basic introduction?

- Requirement Gathering: The first phase of SDLC is requirement gathering from customer. Here we gathered requirement of functional and non-functional requirements by customer
- Analysis: The second phase of SDLC is analysis of gathered requirements from customer. And analyze how these requirements will be accomplished.
- Design: The third phase of SDLC is design phase, on the basis of analysis of gathered requirement. In this phase the software design documents are prepared. There are two kind of design document HLD & LLD
- Implementation: The fourth phase of SDLC is implementation. In this phase developer starts the development by writing the code.
- Testing: The fifth phase of SDLC is testing, testing for giving bug free and quality product to the customer. In testing phase tester will check the application by the help of comparing expected result and actual result.
- Maintenance : The sixth phase of SDLC is maintenance, the maintenance phase comes after deployment of software. The maintenance can be done by 3 techniques: Corrective maintenance, Adaptive maintenance, Preventive maintenance.

15. Explain phases of waterfall model.

- The waterfall model is a classical lifecycle model. There are some phases as below.
 - Requirement collection
 - Analysis
 - Design
 - Implementation
 - Testing
 - Maintenance
- Requirement collection: The aim of this phase is to understand the need of the customer. The team will collect the requirement from customer and also convince the customer if any requirement is not suitable.
- Analysis: The aim of this phase is to understand the exact requirement of customer and document them properly. Both the customer and developer work together so as to document all functions, performance and interfacing requirements of the software.
- Design: This phase aims to transform requirements gathered into a suitable form. It defines overall software architecture with high level and detail design.
- Implementation: During this phase design is implemented. If design documents are ready then coding phase proceeds smoothly.
- Testing : This phase is highly crucial as the quality of the end product is determined by the testing phase. The better output will give the quality product to the customer and low maintenance after deployment.
- Maintenance: Maintenance is the task performed by every user once the software has been delivered to the customer.

16. Write the phases of spiral model

- 1) Planning
- 2) Risk analysis
- 3) Engineering/construct
- 4) Customer evaluation

17. Explain working methodology of agile model and also write pros and cons.

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. The agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continues improvement at every stage.

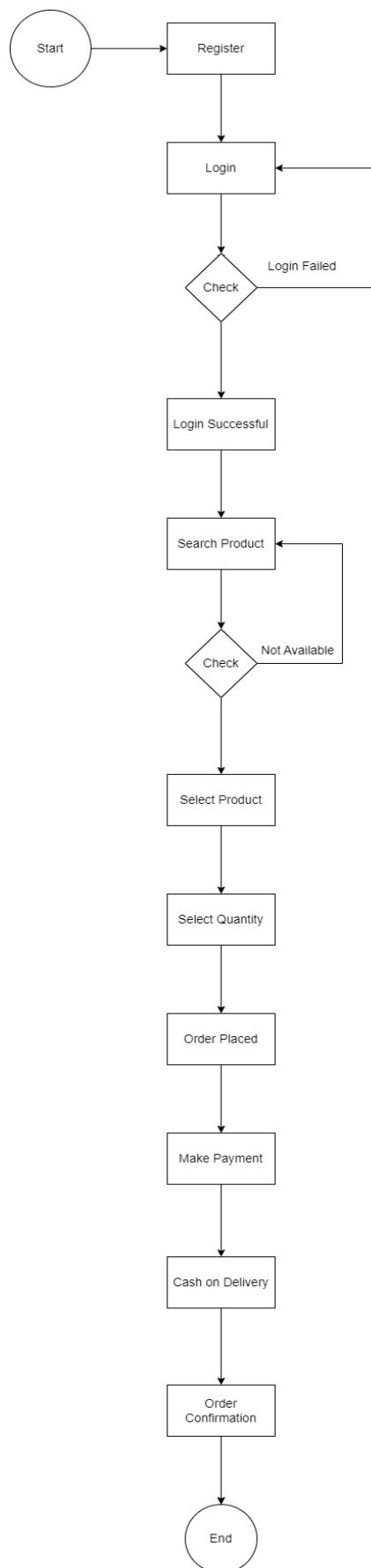
➤ **Pros:**

- It is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fix and changing requirements.
- Delivers early partial working solution
- Good model for environments that change steadily.
- Minimal rules, documentation easily employed.

➤ **Cons:**

- Not suitable for complex dependencies.
- More risk for sustainability, maintainability and extensibility.
- Strict delivery management dictates the scope, functionality to be delivered and adjustments to meet the deadlines.
- Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

18. Draw use case on Online shopping product using COD.



19. Draw use case on Online shopping product using payment gateway.

