**Module -1 Fundamentals**

**1. What is SDLC?**

SDLC is Software development lifecycle. The phases of SDLC are

Planning, Analysis, Design, Build, Testing, Deploy, Maintain.

**2. What is software testing?**

Software Testing is a process used to indentify the correctness , completeness and quality of developed computer software.

It is a process of validating and verifying that a software program or application or product:

- meets the business and technical requirement that guided it's design and development

- works as expected

- can be implemented with the same characteristic

**3. What is Agile methodology?**

Agile methodology focus on process adaptability and customer satisfaction by rapid delivery of working software product. The tasks are divided to time boxes to deliver specific features for a release. In this iterative approach is taken and working software build is delivered after each iteration. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirement analysis, design, coding, unit testing, and acceptance testing. Each build is incremental in terms of features and the final build holds all the features required by customer. example, Google chrome, Mozilla Firefox, Face book.

**4. What is SRS?**

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real- world situation.

**5.what is oops?**

oops is Object oriented programming. It is a programming paradigm based on concept of object.

**6. Write basic concept of oops.**

* Basic concept of oops are:
* Object
* Class
* Encapsulation
* Inheritance
* Polymorphism
* Overriding
* Overloading
* Abstraction

**7. What is Object?**

An object can be defined as a data field that has unique attributes and behavior. An object represent an individual , identifiable item ,unit or entity, either real or abstract, with a well defined role in problem domain. An object is anything to which a concept applies.

**8. What is class?**

Class is a blue print for an object. example: fruit is a class for apple, banana etc and object is an example for a class. A class represents an abstraction of the object and abstract the properties and behavior of that object.

**9. What is encapsulation?**

Encapsulation refers to the wrapping up of data in to a single unit. Any external entity cannot access the data from outside of a class. Data remain hidden.

It describe the idea of bundling data and methods that work on data with one unit and unit here is a class. 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.

**10. What is inheritance?**

Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship. 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 characteristics from a parent class and then add unique features of its own.

For example consider a Vehicle parent class and its child class Car.

Vehicle class will have all common properties and functionalities for all vehicles in common and Car will inherit those common properties from the Vehicle class and then add those properties which are specific to a car. Here, Vehicle is known as base class, parent class, or super class. Car is known as derived class, Child class or subclass.

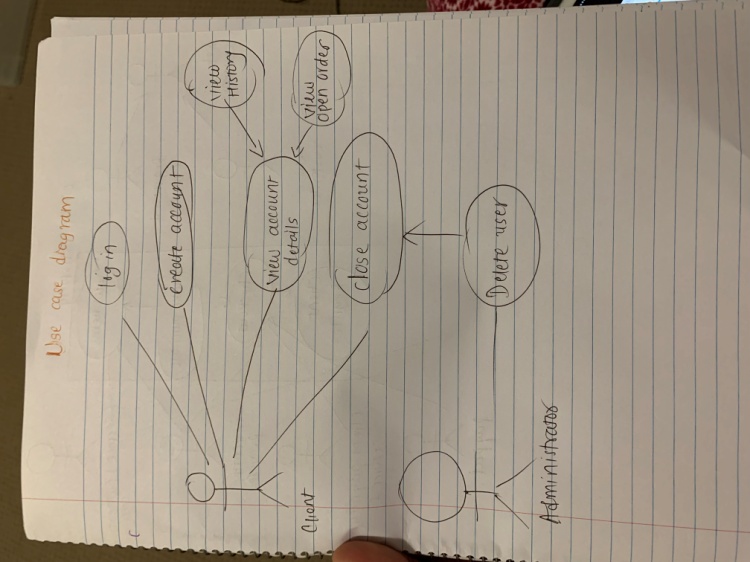
**11. What is polymorphism?**

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.

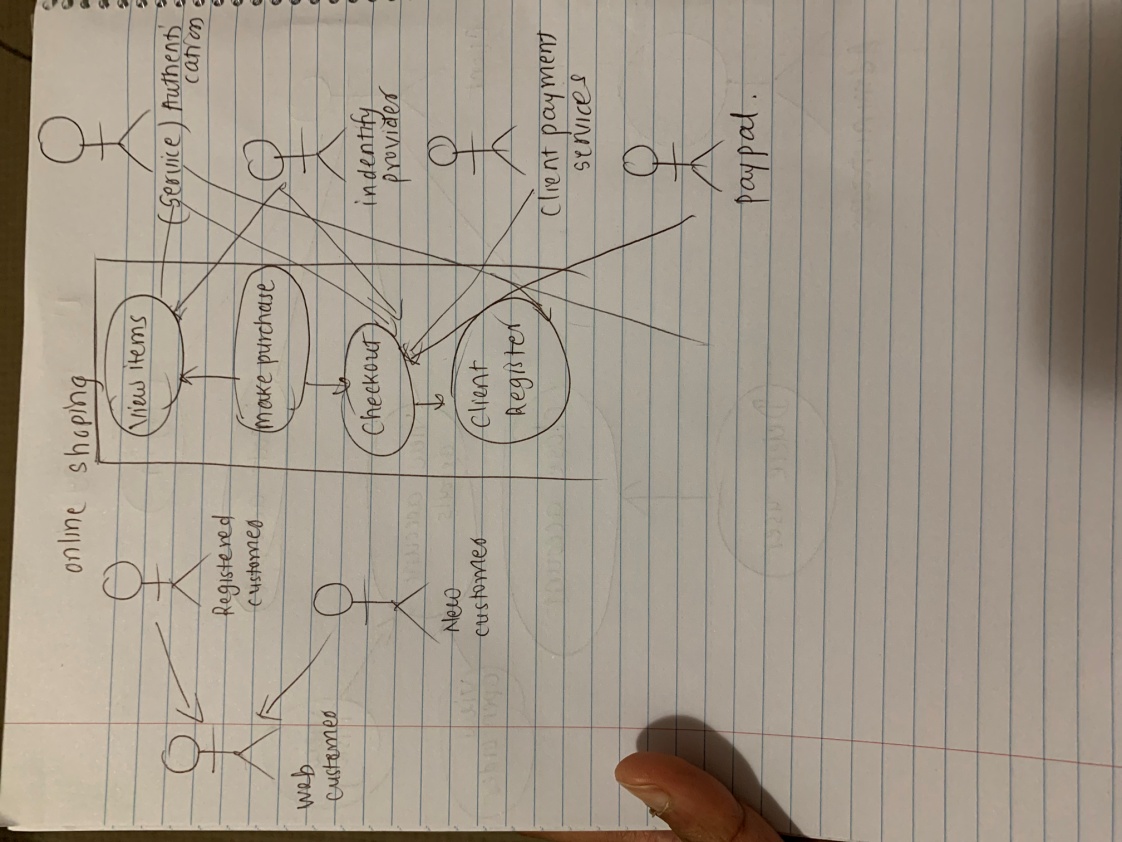
The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism. The ability to change form is known as polymorphism. There is two types of polymorphism in Java:

* Compile time polymorphism(Overloading
* Runtime polymorphism(Overriding)

**12. Draw Use case on Online book shopping.**

****

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



**14. Write SDLC phases with basic introduction.**

SDLC is Software development lifecycle. The phases of SDLC are

Planning, Analysis, Design, Build, Testing, Deploy, Maintain.

* Planning: Find scope, Solution, Estimate cost, Efforts , Identity risk, benefits
* Analysis: Focus on functional requirement. Perform solution analysis, derive end users needs.
* Design: Elaborate core specification, features, functions and flows to meet the requirements.
* Build: Actual development begins here following the design. Code reviews, unit testing are done.
* Testing: QA team tests the software for errors/bugs and determine if the solution meets the business goals.
* Deploy: After successful testing software gets deployed in production environment for user acceptance and approval.
* Maintain: After software passes all the stages, it enters into the maintenance phase. The vendor here provide support for future upgrades and bug fixes.

**15. Explain Phases of the waterfall model.**

* Requirement collection
* Analysis
* Design
* Implementation
* Testing
* Maintenance

**16. Write phases of spiral model.**

* Planning
* Risk Analysis
* Engineering
* Customer Evaluation

**17. Write agile manifesto principles.**

**The four core values of Agile software development as stated by the Agile Manifesto are:**

* individuals and interactions over processes and tools;
* working software over comprehensive documentation;
* customer collaboration over contract negotiation; and.
* responding to change over following a plan.

**18. 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.

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.

At the end of the iteration a working product is displayed to the customer and important stakeholders.

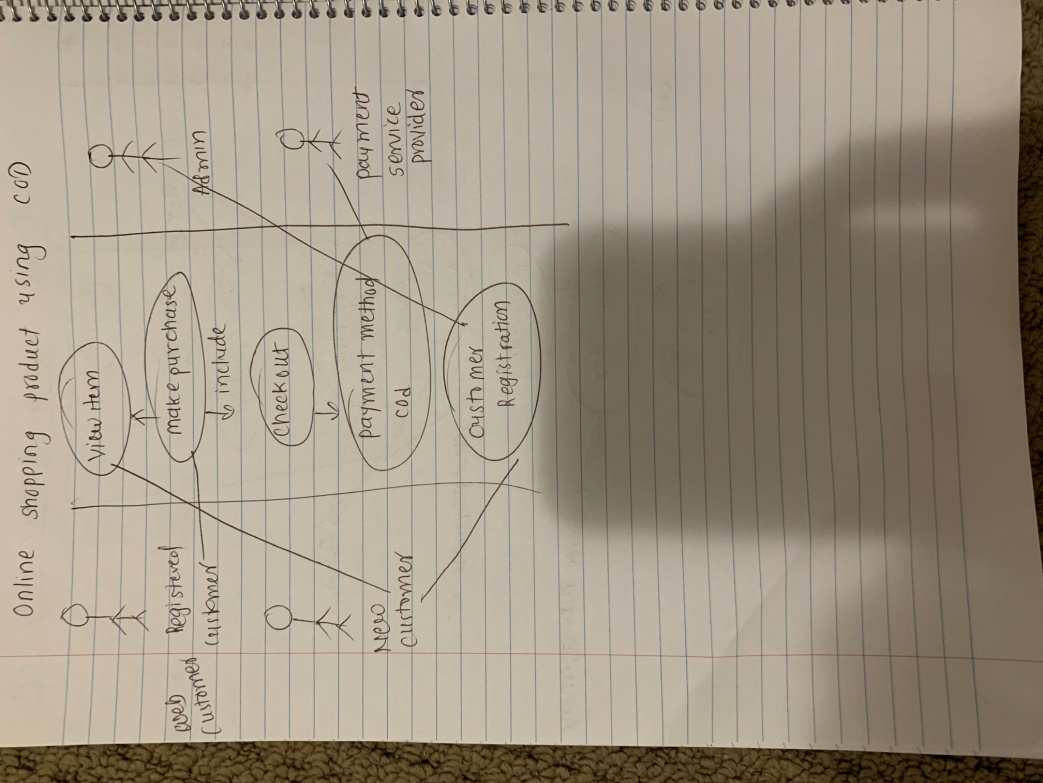
**Pros:**

* 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 fixed or changing requirements
* Delivers early partial working solutions.
* Good model for environments that change steadily.
* Minimal rules, documentation easily employed.
* Enables concurrent development and delivery within an overall planned context.
* Little or no planning required
* Easy to manage
* Gives flexibility to developers

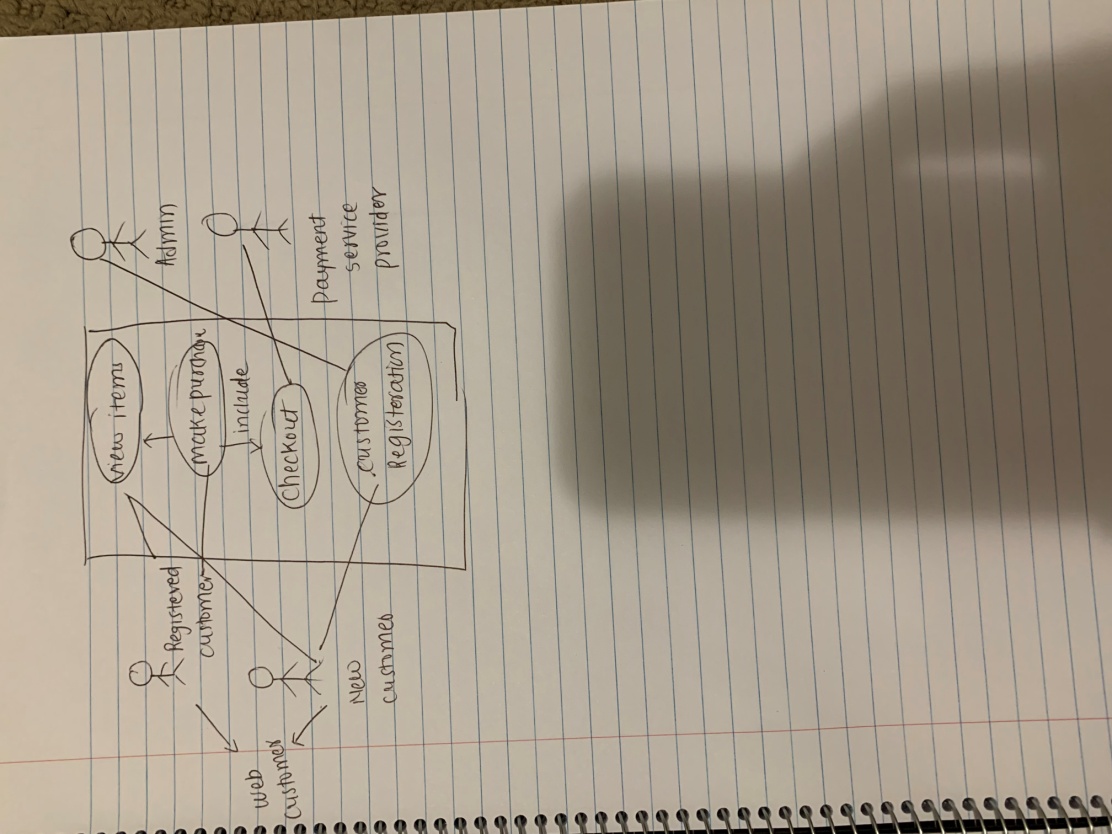
**Cons**

* 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.
* 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.

**19. Draw use case on Online shopping product using COD.**



**20. Draw use case on Online shopping product using payment gateway.**

****