**1. what is SDLC?**

**A.** Software Devlopment Life Cycle(SDLC) is a structure impose on a development of a software product that defines the process for planning,implemention,testing,documentation,deployment,ongoing maintenance nd support.

**2. What is software testing?**

**A.** Software testing is a process used to identify the correctness,completeness,nd quality of complete developed computer software.

**3. What is agile methodology?**

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

Agile method break the product into small incremental builds.

**4. What is SRS?**

**A.** Software Requirement Specification (SRS) is a complete description of behavior of the system to be developed.

**5. What is oops?**

**A.** Object oriented programming has a web of interacting objects each house-keeping its own state.

**6. Write basic concepts of oops?**

**A. 1.** class

**2.** object

**3.** inheritance

**4.** polymorphisam

**5.** encapsulation

**6.** abstraction

**7. What is class?**

**A.** It is a group of different types of variables nd functions.

**8. What is object?**

**A.** It is an instance class.

**9. What is encapsulation?**

**A.** To bind a code nd data into a single unit is called encapsulation.

**10. What is inheritance?**

**A.** The object of one class can aquire the properties of object of another class is called inheritance.

Or

Creating a new class from existing class is called inheritance.

**11. What is polymorphisam?**

**A.** One name multiple form

Types: 1. compile time (method overloading)

When there is more than one method in single class having same

Name but with a different number of arguments and their data

Types is called method overloading.

2. run time (method overriding)

When there is same method prototype in your both base class nd

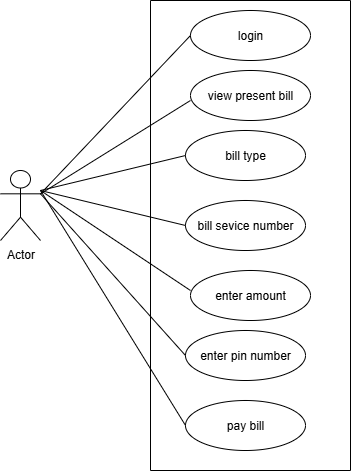
Derived class,if u call that method using the object of derived calss

Nd then only the method of derived class will be call, then un can

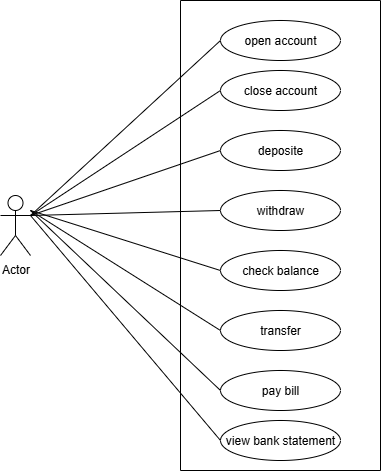
Say that the method of derived class overrides the method of base

Class.

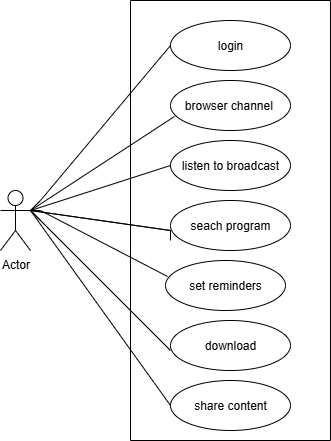
**\*DRAW USECASE ON ONLINE BILL PAYMENT**

****

**\*DRAW USECASE BANKING SYSTEM FOR CUSTOMER**

****

**\*DRAW USECASE BROADCASTING SYSTEM**

****

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

**A. 1.Requirement collection/gathering**

Establish customer needs

**2. Analysis**

Model and specify the requirements-“what”

**3. Design**

model nd specify a solution

**4. Implemantion**

Construct a solution in software

**5. Testing**

Validate the solution against requirements

**6. Maintenance**

Repair defects and adapt the solution to the new requirements.

**13. Explain phases of water fall model?**

**A.** Requirement collection/gathering

Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and table.

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.

**2.** **Analysis**

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.

This phase defines the problem that the customer is trying to solve.

This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.

The architecture defines the components, their interfaces and behaviors

**3.Design**

The design phase of the Waterfall model transforms Software Requirements Specification (SRS) documents into a blueprint for the software product, breaking into logical (abstract, hardware-independent) and physical (detailed, hardware-specific) design stages.

 It establishes the system's architecture, database structure, algorithms, user interface, and required hardware and software.

**4.Implementation**

In the implementation phase, the team builds the components either from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.

**5.Testing**

In the waterfall model, the testing phase is a distinct, sequential stage that occurs after the development phase.

This phase focuses on ensuring the software functions correctly and meets the specified requirements through various testing types like [unit testing](https://www.google.com/search?sca_esv=78726570b3b899ae&sxsrf=AE3TifOIaQIew1fB-yi2aSPWZosAyvWg5A%3A1755780018701&q=unit+testing&sa=X&ved=2ahUKEwji68Wi9puPAxU3SmcHHUwRJCEQxccNegQIIRAB&mstk=AUtExfAzaIzg0eMtdr6ycalC2gvIMr_BJJOCKuhyyYzQ7sCjtW0nEzO-jUyjB6a8HkubEHYGIE2Yf8gKoyOGuQ_DBwrxZ7maNHN_-OJz-ANavW5hc6wZFjrvqYkiP5thKyXsYHdDSEbhURNg3TcmBoIk5G1yXCYGD7fgFRAGMlsdY9PEl0zoo0b-6hQkZWhO2YDdPJxEqnatJKdtyGKrWqwaBPVrdDzKeXu5658BegyEnGKJDwhRWvJxva1t-I2fn7aSAvcf0Zkdh4bWPSEkhM6PBFhE&csui=3), [integration testing](https://www.google.com/search?sca_esv=78726570b3b899ae&sxsrf=AE3TifOIaQIew1fB-yi2aSPWZosAyvWg5A%3A1755780018701&q=integration+testing&sa=X&ved=2ahUKEwji68Wi9puPAxU3SmcHHUwRJCEQxccNegQIIRAC&mstk=AUtExfAzaIzg0eMtdr6ycalC2gvIMr_BJJOCKuhyyYzQ7sCjtW0nEzO-jUyjB6a8HkubEHYGIE2Yf8gKoyOGuQ_DBwrxZ7maNHN_-OJz-ANavW5hc6wZFjrvqYkiP5thKyXsYHdDSEbhURNg3TcmBoIk5G1yXCYGD7fgFRAGMlsdY9PEl0zoo0b-6hQkZWhO2YDdPJxEqnatJKdtyGKrWqwaBPVrdDzKeXu5658BegyEnGKJDwhRWvJxva1t-I2fn7aSAvcf0Zkdh4bWPSEkhM6PBFhE&csui=3), [system testing](https://www.google.com/search?sca_esv=78726570b3b899ae&sxsrf=AE3TifOIaQIew1fB-yi2aSPWZosAyvWg5A%3A1755780018701&q=system+testing&sa=X&ved=2ahUKEwji68Wi9puPAxU3SmcHHUwRJCEQxccNegQIIRAD&mstk=AUtExfAzaIzg0eMtdr6ycalC2gvIMr_BJJOCKuhyyYzQ7sCjtW0nEzO-jUyjB6a8HkubEHYGIE2Yf8gKoyOGuQ_DBwrxZ7maNHN_-OJz-ANavW5hc6wZFjrvqYkiP5thKyXsYHdDSEbhURNg3TcmBoIk5G1yXCYGD7fgFRAGMlsdY9PEl0zoo0b-6hQkZWhO2YDdPJxEqnatJKdtyGKrWqwaBPVrdDzKeXu5658BegyEnGKJDwhRWvJxva1t-I2fn7aSAvcf0Zkdh4bWPSEkhM6PBFhE&csui=3), and [acceptance testing](https://www.google.com/search?sca_esv=78726570b3b899ae&sxsrf=AE3TifOIaQIew1fB-yi2aSPWZosAyvWg5A%3A1755780018701&q=acceptance+testing&sa=X&ved=2ahUKEwji68Wi9puPAxU3SmcHHUwRJCEQxccNegQIIRAE&mstk=AUtExfAzaIzg0eMtdr6ycalC2gvIMr_BJJOCKuhyyYzQ7sCjtW0nEzO-jUyjB6a8HkubEHYGIE2Yf8gKoyOGuQ_DBwrxZ7maNHN_-OJz-ANavW5hc6wZFjrvqYkiP5thKyXsYHdDSEbhURNg3TcmBoIk5G1yXCYGD7fgFRAGMlsdY9PEl0zoo0b-6hQkZWhO2YDdPJxEqnatJKdtyGKrWqwaBPVrdDzKeXu5658BegyEnGKJDwhRWvJxva1t-I2fn7aSAvcf0Zkdh4bWPSEkhM6PBFhE&csui=3).

**6.Maintanance**

The "Maintenance" phase in the Waterfall model is the final stage where the software product or system is actively used by the end-users and ongoing support is provided.

 This phase involves fixing bugs, implementing updates, and providing enhancements based on user feedback or changing business requirements

**14. Write phases of spiral model?**

**A.** 1. Planning

2.risk analysis

3. enginnering or development

4.evaluation

**15. Write agile manifesto principles** .

**A.** Agile Manifesto outlines 12 key principles to guide software development, emphasizing flexibility, collaboration, and customer satisfaction. These principles revolve around delivering working software frequently, adapting to changing requirements, and fostering strong communication and teamwork.

1.customer satisfaction

2.changing requirements

3.frequent delivery

4.communicate regulary

5.face-to-face communication

6.measure work process

7.development process

8.good design

9.measure progress

10.support team member

11.continue seeking result

12.reflect and adjust regulary

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

**A.** 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 .

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

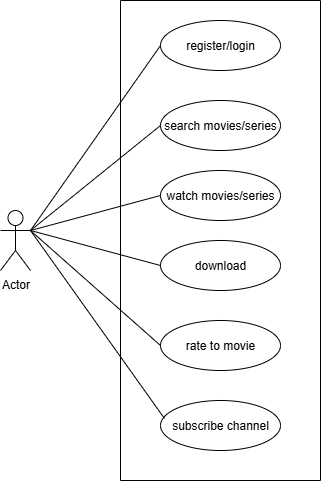
**PROS**

1. Is a very realistic approach to software development Promotes teamwork and cross training.
2. Functionality can be developed rapidly and demonstrated. Resource requirements are minimum
3. Suitable for fixed or changing requirements Delivers early partial working solutions.
4. Gives flexibility to developer
5. Little or no planning required Easy to manage

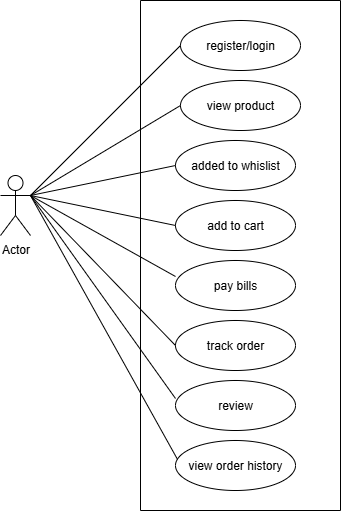
**CONS**

1. Not suitable for handling complex dependencies.
2. More risk of sustainability, maintainability and extensibility.
3. Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
4. There is very high individual dependency, since there is minimum documentation generated.

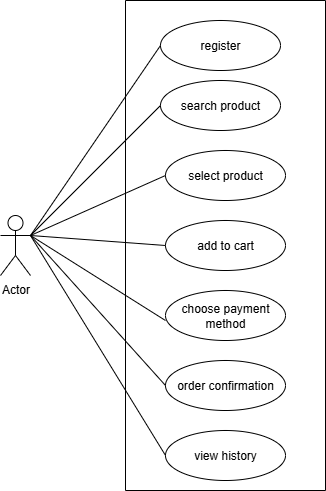
**\*DRAW USECASE ON OTT PLATFORM**

****

**\*DRAW USECASE E-COMMARCE APPLICATION**

****

**\*DRAW USECASE ONLINE SHOPPING USING PAYMENT GATEWAY**

****