Que 1: What is SDLC?

Ans: 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.

Que 2: What is software testing?

Ans: Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

----->Testing apart from finding errors is also used to test performance, safety, fault-tolerance or security.

Que 3: What is agile methodology?

Ans: It is one of the most popular approaches that is used in software development. Agile development is considered as the combination of incremental and iterative work sequences that concentrate on customer satisfaction and process adaptability by delivering the working product quickly.

Que 4: What is SRS?

Ans: A requirement is an action or feature that a product needs to fulfill.

----->All of the specifications that must be applied to the product's design and verification are gathered into a requirement specification.

Que 5: What is oops?

Ans: When creating computer software, programmers can specify both the data type and function types that can be applied to a structure by using an approach known as object-oriented programming.

Que 6: Write Basic Concept of oops.

Ans: **1. Encapsulation:** It is the wrapping of data and associated functions in one single unit.

2. Abstraction: It is the act of representing the essential features without including the background details.

3. Polymorphism: It is the ability for data to be processed in more than one form.

---->Compile time Polymorphism (Operator Overloading)

---->Run time Polymorphism (Operator Overriding)

4. Inheritance: It is the capability of one class to inherit properties from other class.

---->Types of Inheritance:

- Single Inheritance

- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multiple Inheritance

Que 7: What is object?

Ans: Object is a real world entity such as book, fruits etc.

----> An object is created memory is allocated.

Que 8: What is class?

Ans: Class is a blueprint from which objects are created.

---->When a class is created no memory is allocated.

Que 9: What is encapsulation?

Ans: Encapsulation is one of the fundamentals of oop.

---->It describes the combination of data and the operations performed on that data. A structured data object's values or state can be hidden inside a class using encapsulation, limiting direct access by unauthorized parties.

Que 10: What is inheritance?

Ans: One method to depict actual links between the two is through inheritance.

---->As an illustration, consider this: Apples, bananas, and other fruits are all included in the larger fruit category.

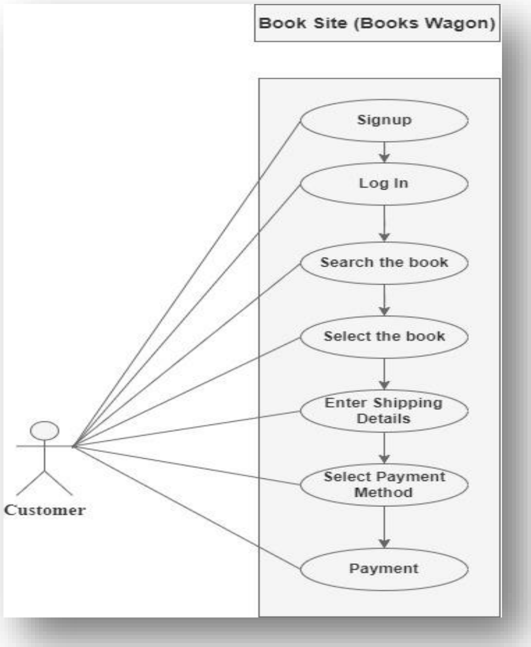
---->It implies that they have inherited the characteristics of the fruit class. Everybody uses them to eat.

Que 11: What is polymorphism?

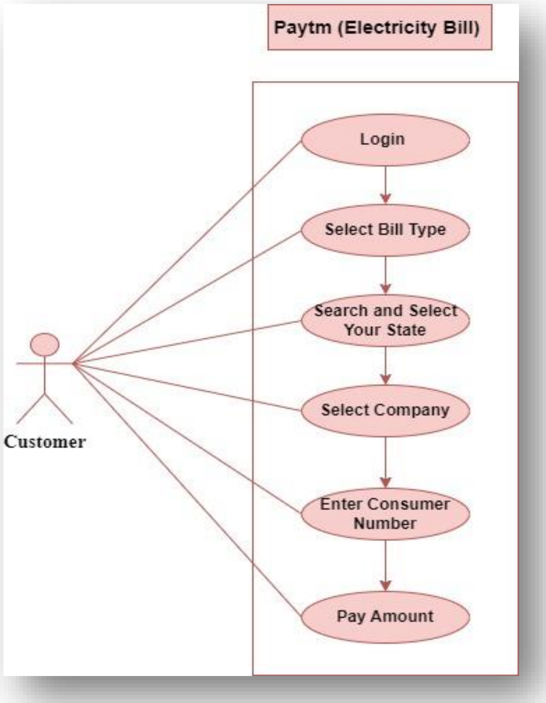
Ans: It describes the ability of something to have or to be displayed in more than one form.

---->Real life example of polymorphism, a person at the same time can have different characteristic. Like a man at the same time is a father, a husband, an employee.

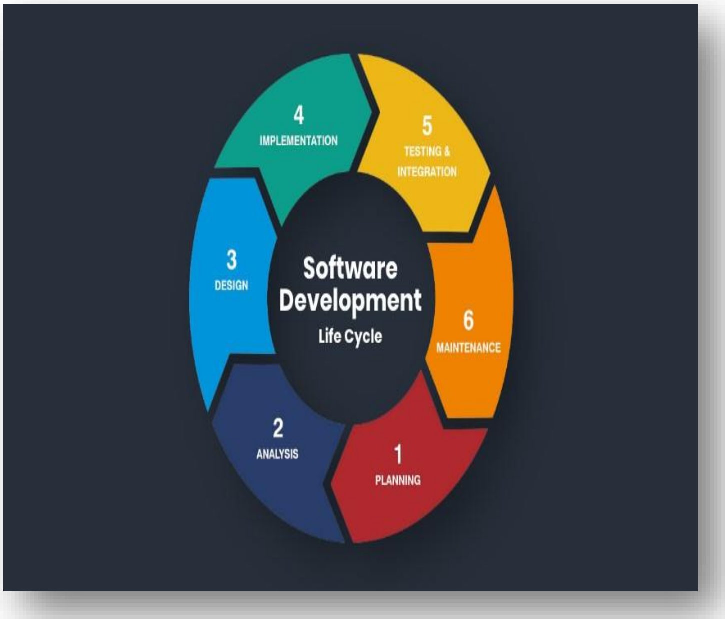
Que 12: Draw Usecase on online book shopping.



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



Que 14: Write SDLC phases with basic introduction.



(Software Development Life Cycle)

Ans: **1. Project Planning**: "What do we want?" is the main focus of the SDLC's first stage.

---->This is the phase where the calculates the cost and specifies the needs for the new software, project planning is essential to the software delivery lifecycle.

2. Analysis: The client requirements for the product.

---->The developer will then analyze the requirements keeping the design and code of the software in mind.

---->The main goal of this stage is that everyone understands even the minute detail of the requirement**.**

3. Designing: The programmer carefully examines if the ready-made software satisfies every need of the user.

---->In addition, the project is financially, practically and technologically possible for the client.

---->The developer chooses the program languages such as Java that are appropriate for the project after determining which design approach is the best.

4. Implementation: The developers will then start building the entire system by writing code using the programming languages they chose.

5. Testing: The program is developed by developers and subsequently implemented in the testing environment.

---->After that, the testing team completes its assessment of the system's overall functionality.

---->Following testing, the QA discovers some errors or defects and informs the developers of them.

---->Once the bugs have been fixed, the developer sends the product to QA for another test.

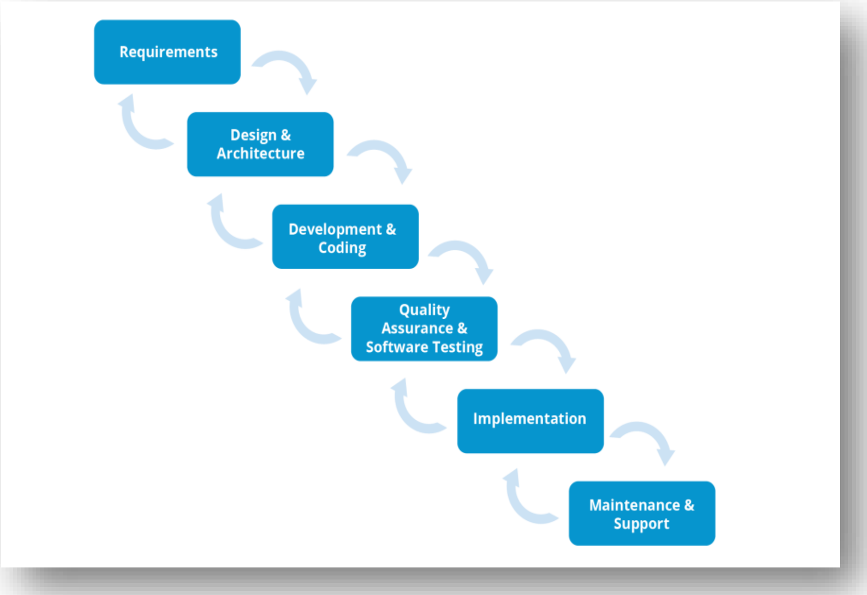
6. Maintenance: When the client utilizes the developed system, the real problem appears, and such problems need to be fixed on a regular basis.

---->Maintenance, the seventh step of the SDLC, is where the developed product is managed.

---->The program is promptly updated to reflect changes in the user's end environment and/or technology.

Que 15: Explain phases of the waterfall model.

Ans:



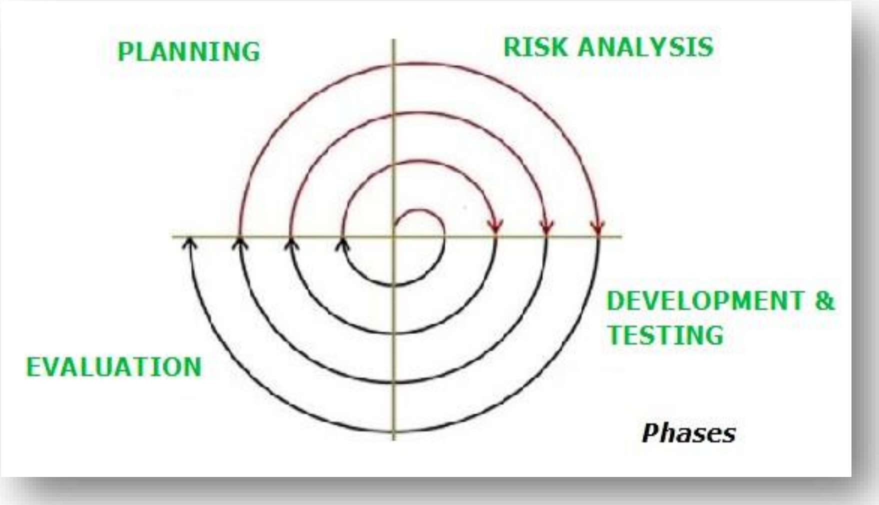
(Waterfall Model)

1. **Requirements**: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

2. **Design:** The system design is prepared which specifies hardware and system requirements, such as data layers, programming languages, network infrastructure etc.
3. **Development:** The product is either released into the market or deployed in the client environment once both functional and non-functional testing is completed.
4. **Testing:** After each unit is tested, all of the units created during the implementation phase are combined into a single system. The entire system is tested for errors and malfunctions after integration.
5. **Implementation:** With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
6. **Maintenance:** In the client setting, some problems can arise. Patches are published to address certain problems. Better versions of the product are also released in an effort to improve it. To implement these modifications in the client environment, maintenance is carried out.

Que 16: Write phases of spiral model.

Ans:



(Spiral Model)

1. **Planning:** Requirements are gathered during the planning phase. Requirements like SRS that is System Requirement specifications.
2. **Risk Analysis:** Possible problems that could impair the project are found during the risk analysis phase. It examines many project components, including resources, requirements, and technology.
3. **Development & Testing:** The program is created and tested with testing done at the end of each phase. Therefore, development and testing are completed at this period.
4. **Evaluation:** Before proceeding to the next stage the client reviews the work completed thus. far in this phase.

Que 17: Write agile manifesto principles.

Ans: 1. Individuals and interactions over processes and tools.
2. Working software over comprehensive documentation.
3. Customer collaboration over contract negotiation.
4. Responding to change over following a plan.

Que 18: Explain working methodology of agile model and also write pros and cons.

Ans: **1. Requirements:** The companies here gather information on customer's requirements and prioritize them based on different factors. It is an initial factor to understand the direction of the project and progress.

2. Planning: In order to deliver the full software package with all of the features, creating a plan is a necessary first step. There is time allotted in the plan for each iteration's creation and delivery.

3. Development: The development team uses quick and frequent iterations to complete the program development.

4. Testing: Software must be tested to ensure that it meets the organization's quality standards and the expectations of the client.

5. Deployment: It is in this case the organization that fulfills the ultimate goal of enabling software utilization.

6. Maintenance: To make sure that clients' needs and expectations are satisfied, the software is continually maintained.

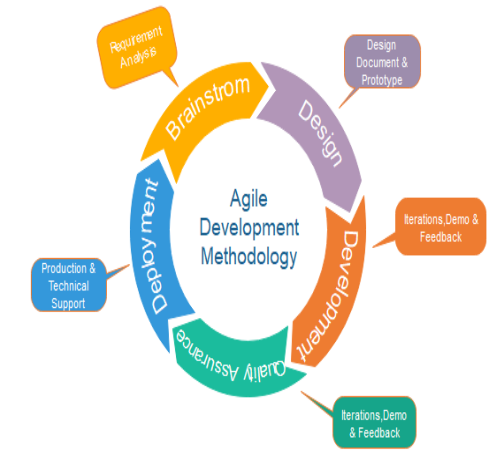

Requirement Analysis
Design Document & Prototype
Brainstrom
Design
Iterations,Demo & Feedback
Agile Development Methodology
Deployment
Development
Production & Technical Support
Quality Assurance
Iterations,Demo & Feedback

Fig. Agile Model

- **Pros:**

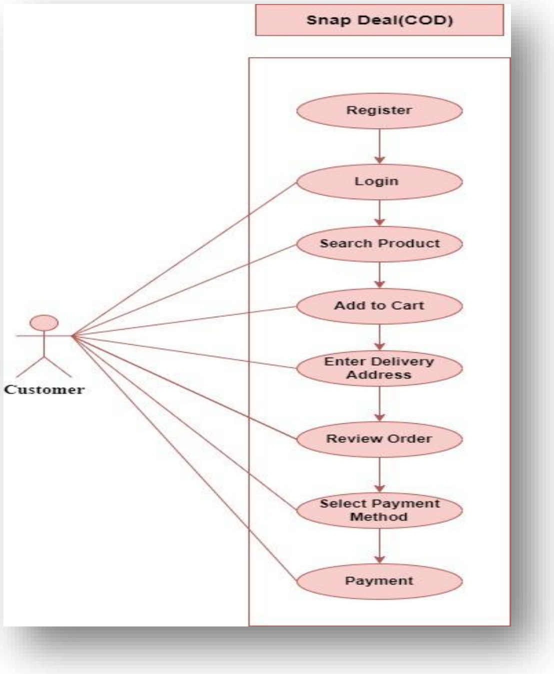
1. Timely delivery
2. Adaptability
3. Ease of collaboration
4. Increased performance improvement
5. Transparency
6. Continuous improvement
7. Higher profits
8. Less preparatory work

- **Cons:**

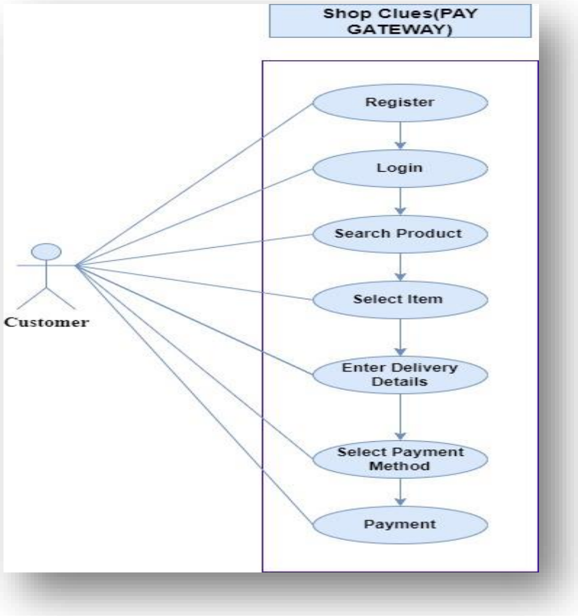
1. Transfer difficulties
2. Variable goals
3. Lack of documentation
4. Less documented improvement
5. Goal focus shifting

6. Less predictability

Que 19: Draw usecase on online shopping product using COD.



Que 20: Draw usecase on online shopping product using payment gateway.



Que 21: What is 7 key principles? Explain in details?

Ans: **1. Testing shows the Presence of Defects:** Software testing aims to break the software. Defects are less common in software when it is tested. Software testing discusses the existence of flaws but not their absence. Software testing can confirm if flaws exist in the program, but it cannot demonstrate that the program is flawless. Software bug-freeness is never guaranteed, not even by numerous tests. While testing can help minimize flaws, it cannot completely eliminate them.

2. Exhaustive Testing is not Possible: Extensive testing is the process of evaluating the software's functionality under all potential input scenarios, including legitimate and invalid ones, as well as prerequisites. Because exhaustive testing is impractical, the program can never be tested across all test scenarios. It can only run a limited number of test cases and makes the assumption that the program is reliable and will generate the right result in each test case. It is not feasible to test every test case in the software, as this would require additional resources such as time and money.

3. Early Testing: Early test activity must be initiated in order to identify the software defect. The cost of a problem found early in the SDLC process will be extremely low. program testing will begin at the beginning, that is, during the requirement analysis process, to improve program performance.

4. Defect Clustering: Most of the flaws in a project can be found in a small number of modules. According to software testing's Pareto Principle, 20% of modules account for 80% of software flaws.

5. Pesticide Paradox: Finding new bugs won't come from repeatedly running the same test cases. To discover new bugs, it is therefore required to examine the test cases and add or update test cases.

6. Testing is Context-Dependent: The software development setting influences the testing methodology. Different software types require different kinds of testing to be done. For instance, testing an Android application is not the same as testing an e-commerce website.

7. Absence of Errors Fallacy: No matter how many bugs are detected and corrected, if your program or system is not useable, it remains unusable**.**

Que 22: Difference between verification and validation.

Ans: **Verification:**

----> Verification is a static practice of verifying documents, design, code and program.

----> It does not involve executing the code.

----> It is human based checking of documents and files.

----> Verification can be achieved by asking “Are you building a product right?”

----> Verification activities are Reviews and Inspections.

Validation:

----> Validation is a dynamic mechanism of validating and testing the actual product.

----> It always involves executing the code.

----> It is computer based execution of program.

----> Validation can be achieved by asking “Are you building a right product?”

----> Validation activity is Testing.