1. What is SDLC?

* “Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares”
* SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process and SDLC methodology focuses on the following phases of software development:

o    Requirement gathering and analysis

o    Design

o    Implementation or coding

o    Testing

o    Deployment

o    Maintenance

2.    What is software testing?

* “A Process of analyzing a software item to detect the difference between existing and required conditions and to evaluate the features of the software item.”
* Software testing is a process of executing a program with the goal of finding errors.

Advantages of software testing:

* Eliminates software bugs, defects, and failures.
* Increases the efficiency of the software.
* Reduces the number of times the software needs to be repaired.
* Improves the quality of the software.
* The testing team can assists the software development team by detecting the mistakes made by them.
* Helps to optimize the code and get rid of unwanted lines of code.
* Prevents future problems, crashes, and complaints from the end users.
* Increases customer satisfaction and user experience.
* Saves time for test execution when automation testing is used.
* Manual testing doesn’t require knowledge of a programming language.
* Test cases and test scenarios that are written for an application may be reusable for other systems.

Disadvantages of Software Testing:

* Most testing types are time-consuming due to executing tests continuously. Thus, the failing tests should repeatedly run until fixing all the issues.
* Lack of experienced software and QA testers who are aware of testing techniques.
* The software testing team requires many members.
* Increases the cost of the software and the budget.
* Enhances the scope and increases the duration of the software development life cycle (SDLC).
* Identifying the exact testing techniques and types (eg: [manual or automation testing](https://accelatest.com/manual-testing-vs-automation-testing-which-is-best-for-your-project/), [black box testing or white box testing](https://accelatest.com/black-and-white-box-testing-ultimate-guide/), etc.) is required depending on the actual requirements.
* Many test management systems are expensive or clunky.

Types of Software Testing Techniques:

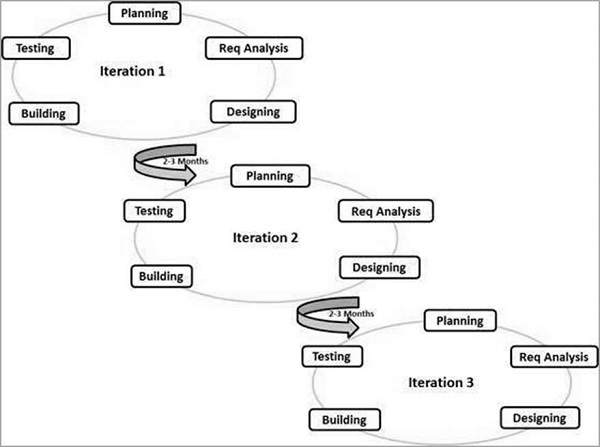
* Functional testing
* Non-Functional testing
* Maintenance Testing.

3.    What is agile methodology?

Write the agile manifesto principle.

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

* Agile Methodology means a practice that promotes **continuous iteration** of development and testing throughout the software development life cycle of the project. In the Agile model in software testing, both development and testing activities are concurrent, unlike the Waterfall model.



* Following are the Agile Manifesto principles −
* **Individuals and interactions** − In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** − Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.
* **Customer collaboration** − As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.
* **Responding to change** − Agile Development is focused on quick responses to change and continuous development.

When to use the agile model?

* When frequent changes are required
* When a highly qualified and experienced team is available
* When a customer is ready to have a meeting with a software team all the time.
* When project size is large.

Advantages of Agile Methodology:

* Customer satisfaction is rapid, continuous development and delivery of useful software.
* Customer, Developer, and Product Owner interact regularly to emphasize rather than processes and tools.
* Product is developed fast and frequently delivered (weeks rather than months.)
* A face-to-face conversation is the best form of communication.
* It continuously gave attention to technical excellence and good design.
* Daily and close cooperation between business people and developers.
* Regular adaptation to changing circumstances.
* Even late changes in requirements are welcomed.

Disadvantages of Agile methodology:

* It is not useful for small development projects.
* There is a lack of intensity on necessary designing and documentation.
* It requires an expert project member to take crucial decisions in the meeting.
* Cost of Agile development methodology is slightly more as compared to other development methodology.
* The project can quickly go off track if the project manager is not clear about requirements and what outcome he/she wants.

4. What is SRS?

* A software requirements specification (SRS) is a document that describes what the software will do and how it will be expected to perform. It also describes the functionality the product needs to fulfill the needs of all stakeholders(business, users).
* Define your product’s purpose
* Describe what you are building
* Detail the requirements
* Deliver it for approval

Qualities | Characteristics of SRS:

* Correct
* Unambiguous
* Complete
* Consistent
* Ranked for importance and/or stability
* Verifiable
* Modifiable
* Traceable
* Types of Requirements: The below diagram depicts the various types of requirements that are captured during SRS.



Advantages of SRS :

* A standard quality of SRS reduces the development cost.
* It should contain all features requested by a client.
* It makes testing easier.
* It defines project scope.

Disadvantages of SRS:

* SRS should not contain any design requirement or design detail.
* It required pictures, tables, charts (bar, worm, etc…).

5. What is oops?

* Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.
* The word object-oriented is the combination of two words i.e. object and oriented. The dictionary meaning of the object is an article or entity that exists in the real world. The meaning of oriented is interested in a particular kind of thing or entity. In layman's terms, it is a programming pattern that rounds around an object or entity and is called object-oriented programming.

Points to Remember:

* Everything is an object
* Developer manipulates objects that uses message passing.
* Every object is an instance of a class.
* The class contains the attribute and behavior associated with an object.

6. Write Basic Concepts of oops

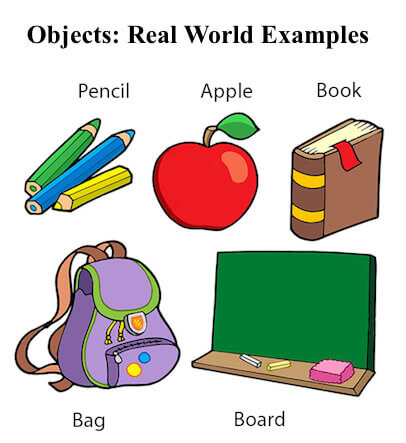
* Classes & Objects
* Abstraction
* Encapsulation
* Inheritance
* Polymorphism

7. What is Object?

* An [Object](https://www.simplilearn.com/tutorials/cpp-tutorial/class-and-object-in-cpp) can be defined as an entity that has a state and behavior, or in other words, anything that exists physically in the world is called an object. It can represent a dog, a person, a table, etc.

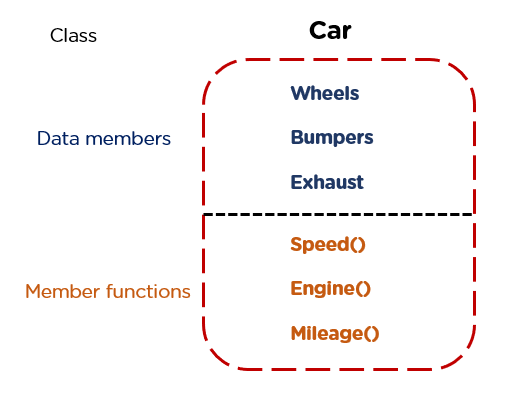
**Object Definitions:**

* An object is a real-world entity.
* An object is a runtime entity.
* The object is an entity which has state and behavior.
* The object is an instance of a class.



8. What is Class?

* [Class](https://www.simplilearn.com/tutorials/cpp-tutorial/classes-in-cpp) can be defined as a blueprint of the object. It is basically a collection of objects which act as building blocks.
* A class contains data members (variables) and member functions. These member functions are used to manipulate the data members inside the class.



A class in Java can contain:

* Fields
* Methods
* Constructors
* Blocks
* Nested class and interface

**Syntax to declare a class:**

class <class\_name>{

    field;

    method;

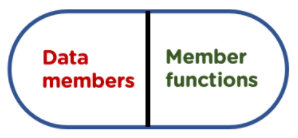
}

9. What is Encapsulation?

* The wrapping up of data and functions together in a single unit is known as encapsulation. It can be achieved by making the data members' scope private and the member function’s scope public to access these data members. Encapsulation makes the data non-accessible to the outside world

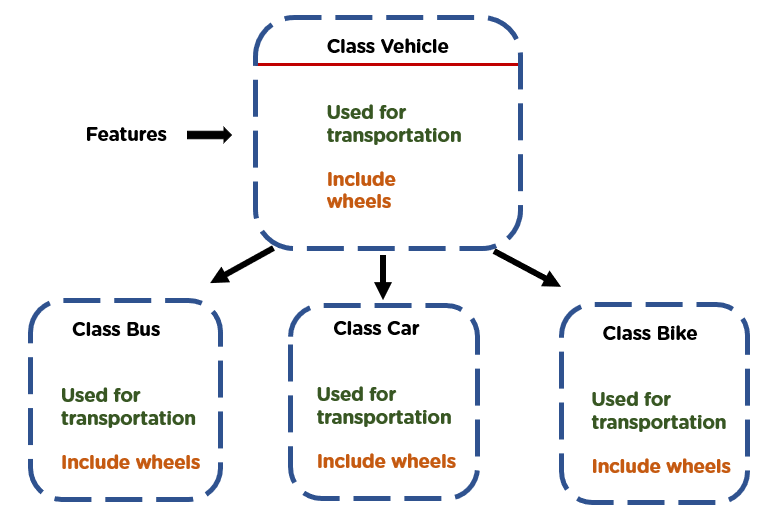
Features of Encapsulation:

* We can not access any function from the class directly. We need an object to access that function that is using the member variables of that class.
* The function which we are making inside the class must use only member variables, only then it is called encapsulation.
* If we don’t make a function inside the class which is using the member variable of the class then we don’t call it encapsulation.
* Increase in the security of data
* It helps to control the modification of our data members.

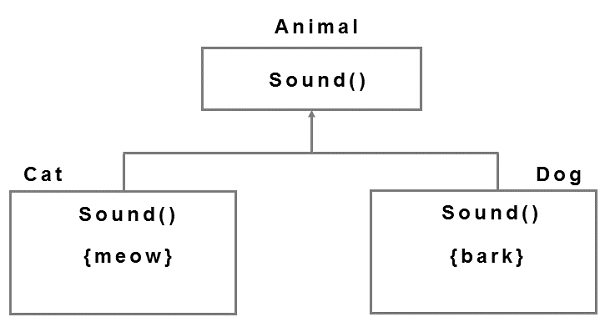


10. What is Inheritance?

* [Inheritance](https://www.simplilearn.com/tutorials/cpp-tutorial/inheritance-in-cpp) is the process in which two classes have an is-a relationship among each other and objects of one class acquire properties and features of the other class. The class which inherits the features is known as the child class, and the class whose features it inherited is called the parent class.For example, Class Vehicle is the parent class, and Class Bus, Car, and Bike are child classes.



11. What is Polymorphism?

* [Polymorphism](https://www.simplilearn.com/tutorials/cpp-tutorial/polymorphism-in-cpp) means many forms. It is the ability to take more than one form. It is a feature that provides a function or an operator with more than one definition. It can be implemented using function overloading, operator overload, [function overriding](https://www.simplilearn.com/tutorials/cpp-tutorial/function-overriding-in-cpp), virtual function.
* Polymorphism is the most essential concept of the Object-Oriented Programming principle. It has the meaning ‘poly’ – many, ‘morph’ – forms.  So polymorphism means many forms.
* In Object-Oriented Programming, any object or method has more than one name associated with it. That is nothing but polymorphism.

12. Online book shop Use case diagram

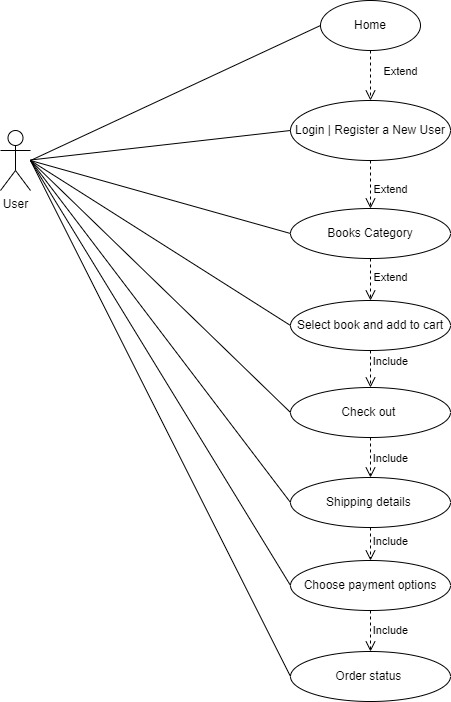


Diagram of Online Book shop

13. Online Bill payment Use case Diagram

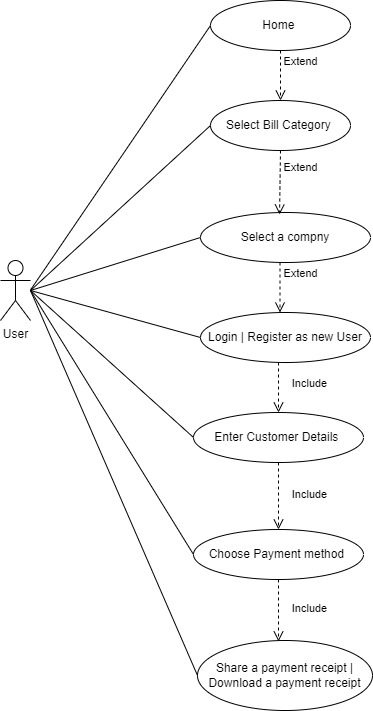


Diagram of Online Bill Payment Use case

14. Write sdlc phases with basic introduction

* SDLC phases are mentioned below:
* Requirement gathering and analysis
* Design
* Implementation or coding
* Testing
* Deployment
* Maintenance

Requirement gathering and Analysis:

* During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.
* Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.
* **For Example,** A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.
* Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

Design:

* In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing system development is derived.

Implementation or Coding:

* Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

Testing:

* Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.
* Retesting, regression testing is done until the point at which the software is as per the customer’s expectation. Testers refer SRS document to make sure that the software is as per the customer’s standard.

Deployment:

* Once the product is tested, it is deployed in the production environment or first [UAT (User Acceptance testing)](https://www.softwaretestinghelp.com/what-is-user-acceptance-testing-uat/) is done depending on the customer expectation.
* In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

Maintenance:

* After the deployment of a product on the production environment, maintenance of the product i.e. if any issue comes up and needs to be fixed or any enhancement is to be done is taken care by the developers.

15. Explain Phases of the Waterfall model:

* **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
* **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.
* **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
* **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

16. Write a phases of Spiral model:

### Identification- This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

* This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

### Design- The Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

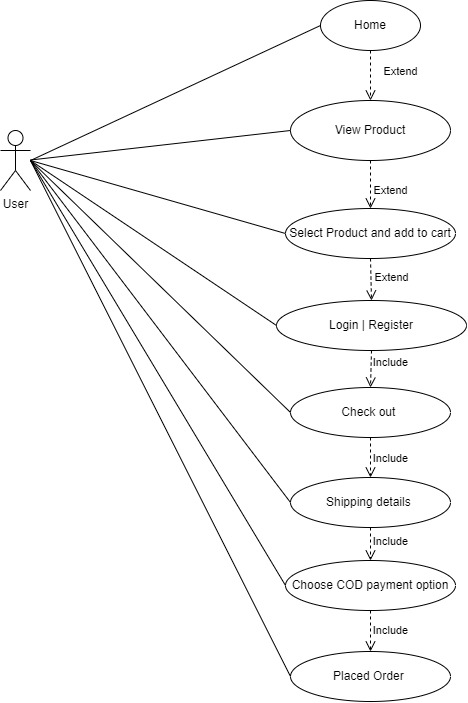
### Construct or Build- The Construct phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

* Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

### Evaluation and Risk Analysis- Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

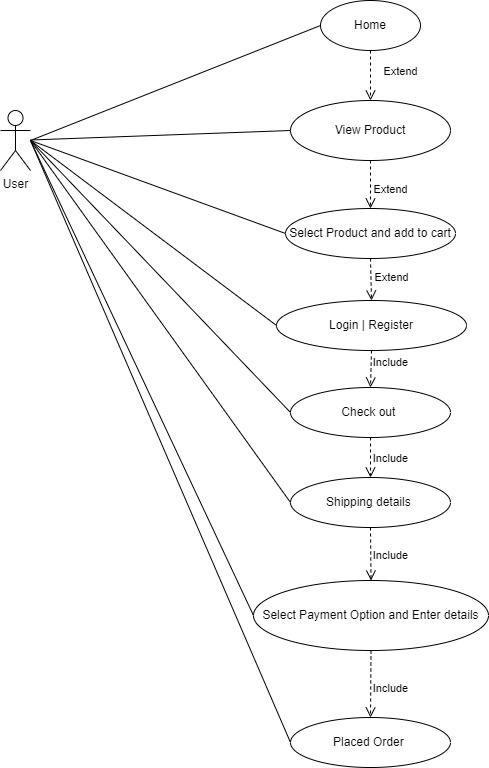
* The following illustration is a representation of the Spiral Model, listing the activities in each phase.

19. Draw usecase on Online shopping using COD.



Online shopping using COD

20. Draw usecase on Online shopping product using payment gateway.

Online shopping using payment gateway