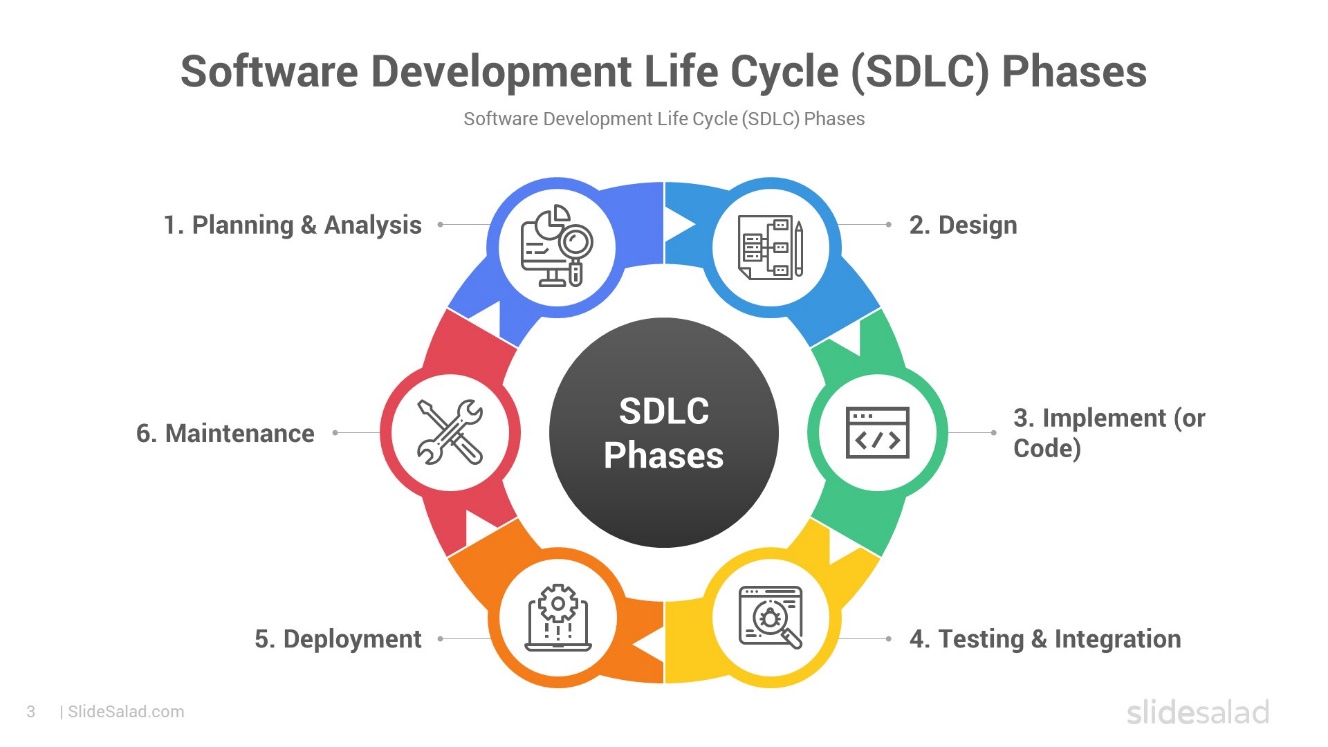
Assignment-1

Module-1 { introduction & fundamental }

QUE-1\_What is software testing ?

* Software testing is the process to verify that the all requirements are fulfilled or not.
* Software testing is the process which is use to identify the currectness, completeness, & quality of the developed software.
* Software testing is the process of evaluating a software product to ensure it meets requirements, works as expected, & is free for defects.
* There are two types of testing :--
* (1) MANUAL TESTING
* (2) AUTOMATION TESTING
* Manual testing :- To execute the test case manually by the test engineer that is called manual testing.
* Automation testing :- To execute the test case by the test engineer with using automation tool that is called automation testing.

QUE-2\_What is SDLC ?



* SDLC :- SOFTWARE DEVELOPMENT LIFE CYCLE.
* SDLC is stand for software development life cycle.
* SDLC is a step by step approach to develop any software / product with high quality, with the time, & within the cost.
* SDLC :- within time + within cost + quality = successful development.
* There are the six phases of SDLC :-
* PLANNING
* ANALYSIS
* DESIGN
* CODING
* TESTING
* MAINTENANCE

QUE-3\_What is SRS ?

* SRS :- Software Requirements Specification.
* SRS is a fully description of the behavior of the system to be developed.
* It is detailed document that outlines the requirements for a software product/system.
* The SRS document provides a comprehensive description of the software’s functional & non-functional requirements.
* Use cases are also known as a functional requirement.
* In addition to use cases , the SRS also contains non-functional requirements.

QUE-4\_Write SDLC phases with basic introduction.

* There are six phases of SDLC.

1. Planning
2. Analysis
3. Design
4. Coding
5. Testing
6. Maintenance
7. Planning :-

* Lack of clarity.
* Requirement confusion (functional / non-functional)
* Requirement group.
* Identify stakeholders & their roles.
* Determine project timeline, budget, &resources.
* Develop a project plan & schedule.

1. Analysis :-

* How the requirements can be executed.
* Gather & document software requirements.
* Develop a details specification document.
* Review & validate requirement with stakeholders.

1. Design :-

* Create a detailed design & architecture.
* Create a detailed design & document.
* Visualize the software/system by designing.
* Like – DFD (Data Flow Daigram), Use Case daigram, ER (entity relationship diagram).
* Review & validate design with stakeholders.

1. Coding :-

* Software implemented by the technology like java, python, php, etc…….
* Write the software code.
* Develop & integrate software components.
* Document code & development process.

1. Testing :-

* Verified that the user requirements fullfill or not.
* All resources are working or not.
* Identify & report, defects & bugs.
* Fix defects & retest software.

1. Maintenance :-

* There are three types of maintenance:-
* Corrective maintenance -- identifying & repair the defects.
* Adaptive maintenance – adapting the existing solution to the new platform.
* Perfective maintenance – implementing the new requirements.

QUE-5\_WHAT IS OOPS ?

* OOPS – OBJECT ORIENTED PROGRAMMING SYSTEM.
* Set of instructions that can be executed by the developer.
* It is a programming paradigm that revolves around the concept of object & classes, which are used to create reusable & modular code.
* OOPS is used in many programming languages, such as JAVA, C++, PYTHON, C#.
* Basic concept of OOPS:-

1. Class
2. Object
3. Encapsulation
4. Inheritance
5. Polymorphism
6. Abstraction

QUE-6\_What is Class?

* A blueprint or template which is collection of data member function.
* A design pattern or template that defines the characteristics & actions of an object.
* Example :-
* Class :- fruit
* Object :- mango

QUE-7\_What is object ?

* An object is an instance of a class, which represents a real-world entity or concept.
* It has its own set of attributes(data) & methods(function) that describe & define its behavior.
* Object will give the memory to the class
* Object will always represent the relavent class.
* Objects interact with each other to perform tasks, making programming more intuitive & modular.

QUE-8\_What is Encapsulation ?

* Encapsulation is a fundamental concept in OOPS that binds together the data& methods that manipulate that data within a single unit, called a class or object.
* Encapsulation in java is the process of wrapping up of members & member functions into single unit.
* It is also hiding the data & implementation details & exposing only the necessary information through control access points.

QUE-9\_What is Inheritance ?

* In OOPS to drive the properties / features / attributes of one class to another class.
* To reusability of code.
* Inheritance describe the relationship between two class.
* Main class :- parent class / base class / super class.
* Another class :- child class / derived class / sub class.

* 5 types of inheritance :-

1. Single inheritance
2. Multilevel inheritance
3. Multiple inheritance
4. Hierarchical inheritance
5. Hybrid inheritance

QUE-10\_What is polymorphism ?

* Polymorphism means having a many forms.
* One name 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.

* 2 types of polymorphism :-

1. Compile time / static binding / overloading.
2. Run time / dynamic binding / overriding.

QUE-11\_Write basic concept of OOPS ?

* OOPS :- Object Oriented programming system.
* Set of instructions that can be executed by the developer.
* 6 types of OOPS :-

1. Class
2. Object
3. Encapsulation
4. Inheritance
5. Polymorphism
6. Abstraction

QUE-12\_Expline phases of waterfall model.



* It is classic software lifecycle models like step “waterfall”, so it is called waterfall model.
* The waterfall model is a traditional & linear approach to software development, where each phases of the project is completed before the next phase begins.
* 6 phases of waterfall model :-
* Gathering :- it describe as gathering information, total cost & time to require to develop any software/product.
* Analysis :- it is short document that define entire lifecycle project.
* Design :- it is a visualization of software of product by designing.
* Coding :- it is a programming code for software.
* Testing :- it is performing if there is no bug in the software or verify the quality, completeness, & correctness of software/product.
* Maintenance :- in simple form, it is sales after service , fixing bugs.

QUE-13\_Write phases of spiral model. 

* The spiral model is a software development life cycle(SDLC) Model that provides a systematic & iterative approach to software development.
* In its diagrammatic representation, looks like a spiral with many loops.
* The exact number of loops of the spiral is unknown & can vary from project to project.
* Each loop of the spiral is called a phase of the software development process.
* Phases of spiral model :-
* Planning :- The next iteration of the spiral begins with a new planning phases, based on the results of the evaluation.
* Risk analysis :- in the risk analysis phase, the risks associated with the project & identified & evaluated.
* Engineering :- in the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
* Evaluation :- in the evaluation phase, the software is evaluated to determine if it meets customer’s requirements & if it is of high quality.

QUE-14\_What is agile methodology ?

* 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 methodology is a structured approach into manageable phases, focusing on continuous improvement.
* It is an iterative process that involves planning, execution, and evaluation.
* Agile Methods break the product into small incremental builds.

QUE-15\_Write agile manifesto principles.

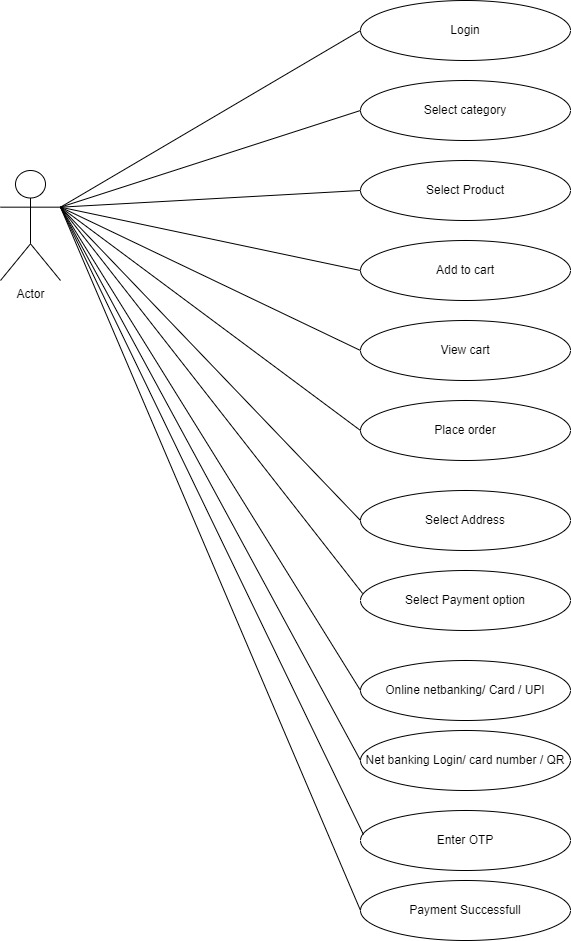
* **Individuals and interactions** :- in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
* **Working software** :- Demo working software is considered the best means of communication with the customer to understand their requirement, 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.

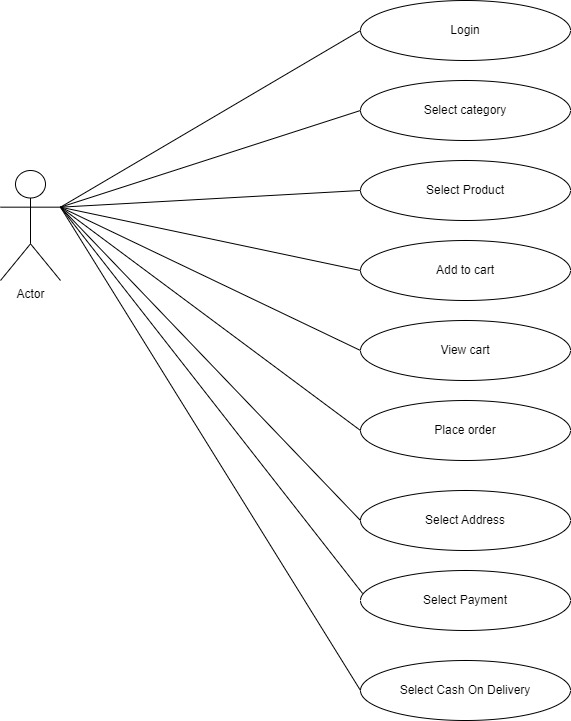
QUE-16\_ Explain working methodology of agile model and, also write pros and cons.

* Agile development model is also a type of Iterative Incremental model.
* Software is developed in incremental, rapid cycles.
* This results in small incremental releases with each release building on previous functionality.
* Each release is thoroughly tested to ensure software quality is maintained.
* It is used for time critical applications.

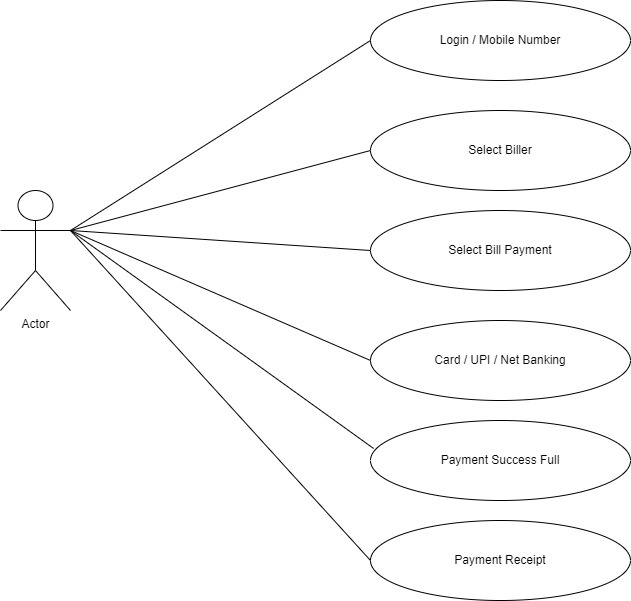
* Pros :-
* It Is a very realistic approach to software development.
* Promotes teamwork and cross training.
* Functionality can be developed rapidly and demonstrated.
* Resource requirements are minimum
* Suitable for fixed or changing requirements.
* Delivers early partial working solutions.
* Good 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.
* There is very high individual dependency, since there is minimum documentation generated.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

QUE-17\_Draw use case on online shopping product using payment gateway.

 QUE-18\_Draw use case on online shopping product using COD.



QUE-19\_Draw use case on online bill payment system (paytm).



QUE-20\_Draw use case on online book shopping.

