**Module 1 (Fundamental)**

1. What is SDLC ?

* SDLC is a structure which is put on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support. There are many different development models.

1. What is software testing ?

* Testing is the process of examine a system or its factors with the intent to find whether it satisfies the specified requirements or not.
* Testing is performing a system in order to identify any gap, errors or missing requirements in contrast to the actual requirements.
* Software is a process used for identify correctness and quality of product.

1. 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 model believes that every project needs to handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

1. What is SRS ?

* A software requirements specification (SRS) is a complete definition of the behavior of the system to be developed.
* This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

1. What is oops ?

* OOPS is Identify objects and assign responsibilities to these objects. Objects communicate to other objects by sending messages. Messages are received by the methods of an object.
* An object is like a black box. The internal details are hidden.
* Object is derived from abstract data type. Object-oriented programming has a web of interacting objects, each house-keeping its own state.
* Objects of a program communicate by sending messages to each other.

1. Write basic concepts of oops.

* Object
* Class
* Encapsulation
* Inheritance
* Polymorphism
* Overriding
* Overloading
* Abstraction

1. What is object ?

* This is the basic unit of object oriented programming(OOP). There are both data and functions that operate on data that are bundled as a unit called as an object.
* Tangible Things as a car, printer, ...
* Roles as employee, boss, ...
* Incidents such as flight, overflow, ...
* Interactions as contract, sale, ...
* Specifications as color, shape, …

1. What is class ?

* Class is a blueprint for an object.
* A class represents an abstraction of the object and abstracts the properties and behavior of that object.
* In the case of a car or laptop, there will be a blueprint or design created first and then the actual car or laptop will be built based on that. We do not actually buy these blueprints but the actual objects.

1. What is Encapsulation ?

* 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.
* Encapsulation enables data hiding, hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.
* Encapsulation is wrapping the things into one capsule.

1. What is inheritance ?

* Inheritance means that one class inherits the characteristics also called a “is a” relationship.
* 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.

Grandparent

Parent

Child

1. 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 most important thing of an object is its behavior. A behavior is initiated by sending a message to the object (usually by calling a method).
* The ability to change form is known as polymorphism.

1. Draw Usecase on Online book shopping.

* <https://drive.google.com/file/d/1dIzefQY-v4RBaI01BiuqHJw5f8bJHH2M/view?usp=sharing>

1. Draw Usecase on online bill payment system.

* <https://drive.google.com/file/d/1dIzefQY-v4RBaI01BiuqHJw5f8bJHH2M/view?usp=sharing>

1. Write SDLC phases with basic introduction.

* There are six phases of SDLC.

✓ Requirements collection/ gathering:

* Establish customer needs
* Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
* Usage scenarios
* Types of requirements:

1 Functional Requirements: describe system services or functions.

2 Non-Functional Requirements are constraints on the system or the development process.

✓ Analysis Phase:

* The analysis phase defines the requirements of the system, independent of how these

requirements will be accomplished.

* Ideally, this document states in a clear and precise fashion what is to be built.

✓ Design Phase:

* Design Architecture Document
* Implementation Plan
* Critical Priority Analysis
* Performance Analysis
* Test Plan

✓ Implementation Phase:

* 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.
* The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.
* The end deliverable is the product itself. There are already many established techniques associated with implementation.

✓ Testing phase :

* Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.
* It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.
* A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.
* Quality is a distinguishing attribute of a system indicating the degree of excellence.

✓ Maintenance phase :

* The maintenance phase is the phase which comes after deployment of the software into the field.
* The developing organization or team will have some mechanism to document and track defects and deficiencies.
* reengineering (redesigning and refactoring)
* updating all analysis, design and user documentation
* Repeatable, automated tests enable evolution and refactoring
* Maintenance is the process of changing a system after it has been deployed.

1. Explain Phases of the waterfall model.

* The waterfall is unrealistic for many reasons, especially:
* Requirements must be “frozen” to early in the life cycle
* Requirements are validated too late.

**Phases :**

1. Requirements collection
2. Analysis
3. Design
4. Implementation
5. Testing
6. Maintenance

16. Write phases of spiral model.

* Spiral Model is very widely used in the software industry as it is in sync with the natural development process of any product i.e. learning with maturity and also involves minimum risk.
* There are 4 phases of the spiral model.

1. Planning : Determination of objectives, alternatives and constraints.
2. Risk analysis : Analysis of alternatives and identification/ resolution of risks.
3. Engineering : Development of the next level product
4. Customer evaluation: Assessment of the result of engineering

17. Write agile manifesto principles.

* 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 products.

✓ Agile Manifesto Principles

1. Individuals and interactions - in agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.
2. 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.
3. 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..

1. Responding to change - agile development is focused on quick responses to change and

continuous development.

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.
* The Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are

divided into time boxes (small time frames) to deliver specific features for a release.

**✓ 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.
* 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 the 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

19. Draw Usecase on online shopping product using COD.

* <https://drive.google.com/file/d/17VEErHnBr8cLlFMgtMcstRmLNast-k1S/view?usp=sharing>

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

* <https://drive.google.com/file/d/1rl7PdIn35bA3kIT8MBBLlTGOMdzKMeBu/view?usp=sharing>