Module-1(fundamental)

1. What is SDLC

-> SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.

-> SDLC phase are:

* Requirements Gathering: in this phase we collect the requirement of client in document format.
* Analysis: in this phase we analysis the document and market requirement.
* Design: in this phase we do the front end part(design of project or product).
* Implementation: in this phase we can do the codding part of this for example in which language we work in, in which database we work on and than which server we implement the project.
* Testing: in this phase we do the testing of the project or product for example unit testing, integrated testing, back box etc.
* Maintenance: in this phase we can maintain the project after deployment.

1. What is Software testing?

-> Software testing is a process that use to identify the completeness, correctness and quality of developed computer software.

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 Methods break the product into small incremental builds. These builds are provided in iterations.

Each iteration typically lasts from about one to three weeks.

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

Is a very realistic approach to software development Promotes teamwork and cross training.

1. What is SRS?

-> A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.

It includes both functional and non-functional requirements, and it serves as a contract between the client and the development team.

It also includes a set of use cases that describe all of the interactions that the users will have with the software

Non-functional requirements are requirements which impose constraints on the design or implementation.

1. What is oops?

-> Identifying objects and assigning 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 interact by sending messages to each other.

-> Concepts of OO

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

1. Write Basic Concepts of oops

-> there are four main concepts of oops :

* Encapsulation
* Abstraction
* Inheritance
* Polymorphism
* Encapsulation: Wrapping data or variables and methods together as a single unit (class). It Hides the internal details of an object and only exposes necessary parts.
* Abstraction: Showing only the essential features of an object and hiding the complex details. Simplifies code and reduces complexity.
* Inheritance: One class (child/subclass) inherits properties and behavior from another class (parent/superclass).
* Polymorphism: The ability of one function or method to behave differently based on the object.

1. What is object

-> An object is a real-world entity or a specific instance of a class in programming, especially in Object-Oriented Programming (OOP).

The two part of an object:

Object = Data + Methods

1. What is class

-> A class is a blueprint or template used to create objects in object-oriented programming.

This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

A class represents an abstraction of the object and abstracts the properties and behavior of that object.

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 is placing the data and the functions that work on that data in the same place. While working with procedural languages, it is not always clear which functions work on which variables but object- oriented programming provides you framework to place the data and the relevant functions together in the same object.

Encapsulation in Java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit; and the unit here is a Class (or interface).

1. What is inheritance

-> Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.

This is a very important concept of object-oriented programming since this feature helps to reduce the code size.

In general, Java supports single-parent, multiple-children inheritance and multilevel inheritance (Grandparent-> Parent -> Child) for classes and interfaces. Java supports multiple inheritances (multiple parents, single child) only through interfaces.

In a class context, inheritance is referred to as implementation inheritance, and in an interface context, it is also referred to as interface inheritance.

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 aspect of an object is its behaviour (the things it can do). A behaviour is initiated by sending a message to the object (usually by calling a method).

The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism.

1. Draw Usecase on Online book shopping

-> https://drive.google.com/file/d/1IMXxvb2QvEf\_nM9HIQLfqptAfN4sS8wv/view?usp=sharing

1. Draw Usecase on online bill payment system (paytm)

-> https://drive.google.com/file/d/1dKWqfpvWiPOaFnpv5ddvYftAH4YmqB4G/view?usp=sharing

1. Write SDLC phases with basic introduction.

-> SDLC phases:

* Requirements Collection/Gathering
* Analysis
* Design
* Implementation
* Testing
* Maintenance
* Requirements Collection/Gathering:
* Features
* Usage scenarios
* Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
* Requirements will Change!
* 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 analysis represents the “what” phase.
* Design:
* Design Architecture Document Implementation Plan
* Critical Priority Analysis Performance Analysis Test Plan
* The Design team can now expand upon the information established in the requirement document.
* The requirement document must guide this decision process.
* 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.
* For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reus ability guideline.
* Testing:
* Simply stated, quality is very important. 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
* Maintenance:
* Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects
* Maintenance is the process of changing a system after it has been deployed.
* Corrective maintenance: identifying and repairing defects Adaptive maintenance: adapting the existing solution to the new platforms.
* Perfective Maintenance: implementing the new requirements
* In a spiral lifecycle, everything after the delivery and deployment of the first prototype can be considered “maintenance”!

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

Requirements are very well documented, clear and fixed. Product definition is stable.

Technology is understood and is not dynamic. There are no ambiguous requirements.

Ample resources with required expertise are available to support the product.

The project is short.

1. Write phases of spiral model

-> Hear are phases of spiral model:

* Planning: initial requirements.
* Risk analysis: something that will delay project or increase its cost.
* Engineering: Development of the next level product.
* Customer Evaluation: Assessment of the result engineering.

1. Write agile manifesto principles

-> Here are the 12 Principles of the Agile Manifesto:

* Customer satisfaction through early and continuous delivery of valuable software.
* Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
* Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
* Business people and developers must work together daily throughout the project.
* Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
* The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
* Working software is the primary measure of progress.
* Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
* Continuous attention to technical excellence and good design enhances agility.
* Simplicity—the art of maximizing the amount of work not done—is essential.
* The best architectures, requirements, and designs emerge from self-organizing teams.
* At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

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

-> agile working methodology:

* Iterative Development: Work is divided into small units called sprints (usually 1–4 weeks).
* Planning Before Each Sprint: The team and stakeholders plan tasks based on priorities.
* Daily Stand-up Meetings: Short daily meetings to discuss progress, issues, and plans.
* Continuous Feedback: Clients give feedback at the end of each sprint.
* Working Software Delivery: A working product is delivered at the end of each sprint.
* Testing During Development: QA and testing are part of every sprint.
* Team Collaboration: Developers, testers, and stakeholders collaborate closely.
* Review & Retrospective: After each sprint, the team reflects and makes improvements.

-> 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.

-> 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.

1. Draw usecase on Online shopping product using COD.

-> https://drive.google.com/file/d/1SYvvknBWlCRukVeoPCC1IiKsytLGaeoa/view?usp=sharing

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

-> https://drive.google.com/file/d/1KaxBJd1k0sIF8MVDBoSBrh6mFRzcuk1d/view?usp=sharing