Assingment-1 module 1 fundamental

Que-1	**Whats is SDLC?**
Ans-1	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. There are a number of different development models. A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or piece of software.
Que-2	**What is Software Testing?**
Ans-2	1.The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects. 2.Testing is the process of evaluating a system or its component(s) with the intent to find that whether it is satisfies the specified requirements or not. 3.This activity results in the actual, expected and difference between their results. In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirement. 4. Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
Que-3	**What is agile methodology?**
Ans-3	1.The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage Once the work begins, teams cycle through a process of planning, executing, and evaluating. 2.Agile is based on the iterative-incremental model. In an incremental model, we create the system in increments, where each increment is developed and tested individually. 3.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. 4.Agile Methods break the product into small incremental builds.These builds are provided in iterations. 5.Each iteration typically lasts from about one to three weeks. 6.Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing. 7.At the end of the iteration a working product is displayed to the customer and important stakeholders.
Que-4	**What is SRS?**
Ans-4	A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).
Que-5	**What is OOPS?**

Ans-5	1.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. 2.Object Oriented Programming is a methodology or paradigm to design a program using classes and objects. 3.It simplifies the software development and maintenance by providing some concepts:
Que-6	**Wrtie Basic Concept of OOPS?**
Ans-6	1.Objects contain data, referred to as attributes or properties, and methods. 2.Class is collection of Objects. It's logical Entity. OOP allows objects to interact with each other using four basic principles: encapsulation, inheritance, polymorphism, and abstraction. These four OOP principles enable objects to communicate and collaborate to create powerful applications.
Que-7	**What is Object?**
Ans-7	Any entity that has state(data) and behavior(code) ,identity is known as an object. For example chair, pen, table, keyboard, bike etc. It can be physical and logical. An object is an instance of a Class. It contains properties and functions. They are like real-world objects. For example, your car, house, laptop, etc. Each object's name, or identity, is unique and distinct from other objects. State refers to the properties of an object. For example, values of variables in the object contain data that can be added, changed or deleted. Behavior refers to actions that the object can take.
Que-8	**What is Class?**
Ans-8	1.Collection of objects is called class. It is a logical entity.When you define a class, you define a blueprint for an object. 2.A class in Java is a set of objects which shares common characteristics/ behavior and common properties/ attributes. It is a user-defined blueprint or prototype from which objects are created. For example, Student is a class while a particular student named Ravi is an object. 3.When you define a class, you define a blueprint for an object.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. 4.A class represents an abstraction of the object and abstracts the properties and behavior of that object. 5.Class can be considered as the blueprint or definition or a template for an object and describes the properties and behavior of that object, but without any actual existence. 6.An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class. Ex :-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.
Que-9	**What is encapsulation?**

Ans-9	1.Binding (or wrapping) code and data together into a single unit is known as encapsulation For example capsule, it is wrapped with different medicines A java class is the example of encapsulation Java bean is the fully encapsulated class because all the data members are private here. 2. 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. 3.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). 4.Encapsulation enables data hiding, hiding irrelevant information from the users of a class and exposing only the relevant details required by the user. 5. We can expose our operations hiding the details of what is needed to perform that operation.We can protect the internal state of an object by hiding its attributes from the outside world (by making it private), and then exposing them through setter and getter methods. Now modifications to the object internals are only controlled through these methods.
Extra	**What is abstraction?**
	Abstraction is the representation of the essential features of an object. These are ‘encapsulated’ into an abstract data type. 1. Data abstraction refers to, providing only essential information to the outside word and hiding their background details, i.e., to represent the needed information in program without presenting the details. For example, a database system hides certain details of how data is stored and created and maintained. 2.Similar way, C++ classes provides different methods to the outside world without giving internal detail about those methods and data. 3. In plain English, abstract means a concept or idea not associated with any specific instance and does not have a concrete existence. 4. Abstraction in Object Oriented Programming refers to the ability to make a class abstract. 5.Abstraction captures only those details about an object that are relevant to the current perspective. 6.Abstraction tries to reduce and factor out details so that the programmer can focus on a few concepts at a time. Java provides interfaces and abstract classes for describing abstract types.
Que-10	**What is inheritance?**
Ans-10	That enables a new class to inherit the properties and methods of an existing class. This feature promotes code reusability and creates a parent-child relationship between classes. Inheritance is a feature or a process in which, new classes are created from the existing classes. The new class created is called “derived class” or “child class” and the existing class is known as the “base class” or “parent class”. The derived class now is said to be inherited from the base class. Inheritance in Java is the method to create a hierarchy between classes by inheriting from other classes. Java Inheritance is transitive - so if Sedan extends Car and Car extends Vehicle, then Sedan is also inherited from the Vehicle class. The Vehicle becomes the superclass of both Car and Sedan. Inheritance is a mechanism where a new class is derived from an existing class, inheriting its properties and methods. Inheritance means that one class inherits the characteristics ofanother class. This is also called a “is a” relationship.
Que-11	**What is polymorphism?**

Ans-11	1.Polymorphism means "many forms". 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. 2.It allows different objects to respond to the same message in different ways, the response specific to the type of the object. 3.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). 4. It is allows different objects of different classes to be treated as objects of a common super class, primarily through the use of interfaces and abstract classes. 5.Polymorphism in Java is a concept that allows objects of different classes to be treated as objects of a common class. 6.Polymorphism allows us to perform a single action in different ways. In other words, polymorphism allows you to define one interface and have multiple implementations. 7.Poly refers to many. That is a single function or an operator functioning in many ways different upon the usage is called polymorphism. The ability to change form is known as polymorphism. => There is two types of polymorphism in Java 1. Compile time polymorphism(Overloading) 2. Runtime polymorphism(Overriding) 1. Compile time polymorphism(Overloading) a.The concept of overloading is also a branch of polymorphism. When the exiting operator or function is made to operate on new data type, it is said to be overloaded. b.The same method name (method overloading) or operator symbol (operator overloading) can be used in different contents. c.In method overloading, multiple methods having same name can appear in a class, but with different signature. d. And based on the number and type of arguments we provide while calling the method, the correct method will be called. e. Java doesn't allow operator overloading yet + is overloaded for class String. The ‘+’ operator can be used for addition as well as string concatenation.
Que-12	**Draw Usecase on Online Book Shopping?**
Ans-12	As Per Mentor say create one only Online Shopping UseCase Diagram
Que-13	**Draw Usecase on Online Bill Payment System (Paytm)?**
Ans-13	As Per Mentor say create one only Online Shopping UseCase Diagram
Que-14	**Write SDLC Phases With Basic Introduction?**
Ans-14	1. Requirements Gathering :- Establish Customer Needs 2.Analysis :- Model and Specify the requirements -"What". 3.Design :- Model and Specify a Solution - "Why", 4. Implementation :- Construct a Solution In Software. 5.Testing :- Validate the Solution against the requirements. 6. Maintenance :- Repair defects and adapt the solution to the new requirements.
Que-15	**Explain Phases of the Waterfall Model?**

Ans-15	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.
Que-16	**Write Phase of Spiral Model?**
Ans-16	Planning, Risk Analysis, Engineering, Customer Evaluation
Que-17	**Write Agile Manifesto Principles?**
Ans-17	1. Customer satisfaction through early and continuous software delivery – Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases. 2.Accommodate changing requirements throughout the development process – The ability to avoid delays when a requirement or feature request changes. 3.Frequent delivery of working software – Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software. 4.Collaboration between the business stakeholders and developers throughout the project – Better decisions are made when the business and technical team are aligned. 5.Support, trust, and motivate the people involved – Motivated teams are more likely to deliver their best work than unhappy teams. 6.Enable face-to-face interactions – Communication is more successful when development teams are co-located. 7.Working software is the primary measure of progress – Delivering functional software to the customer is the ultimate factor that measures progress. 8.Agile processes to support a consistent development pace – Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release. 9.Attention to technical detail and design enhances agility – The right skills and gooddesign ensures the team can maintain the pace, constantly improve the product, and sustain change. 10.Simplicity – Develop just enough to get the job done for right now. 11.Self-organizing teams encourage great architectures, requirements, and designs –Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products. 12.Regular reflections on how to become more effective – Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.
Que-18	**Explain Working Methodology of Agile Model and also write Pros and Cons.?**

Ans-18	Working 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. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis,design, coding, unit testing, and acceptance testing. At the end of the iteration a working product is displayed to the customer and important stakeholders. Pros:- 1.Is a very realistic approach to software development Promotes teamwork and cross training. 2.Functionality can be developed rapidly and demonstrated. 3.Resource requirements are minimum. 4.Suitable for fixed or changing requirements 5.Delivers early partial working solutions. 6.Good model for environments that change steadily. 7.Minimal rules, documentation easily employed. 8.Enables concurrent development and delivery within an overall planned context. 9.Little or no planning required 10.Easy to manage 11.Gives flexibility to developers Cons:- 1.Not suitable for handling complex dependencies. 2.More risk of sustainability, maintainability and extensibility. 3.An overall plan, an agile leader and agile PM practice is a must without which it will not work. 4.Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines. 5.Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction. 6.There is very high individual dependency, since there is minimum documentation generated. 7.Transfer of technology to new team members may be quite challenging due to lack of documentation.
Que-19	**Draw usecase on Online shopping product using COD?**
Ans-19	As Per Mentor say create one only Online Shopping UseCase Diagram
Que-20	**Draw usecase on Online shopping product using payment gateway?**
Ans-20	As Per Mentor say create one only Online Shopping UseCase Diagram