NAME :- Patel Keval chandrakantbhai

Module :- 1 [ Fundamentals ]

|  |
| --- |
| 1)  **What is SDLC** |
| ANS:- SDLS 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. |
|  |
| The methodology within the SDLC process can vary across industries |
| and organizations, but standards such as ISO/IEC 12207 represent |
| processes that establish a lifecycle for software, and provide a |
| Development, acquisition, and configuration of software systems. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 2) What is software Testing? |
| ANS:- Software Testing is a process used to identify the correctness, |
| Completeness, and quality of developed computer software. |
|  |
| Test execution is only a part of testing, but not all of the testing |
| activities Test activities exist before and after test execution It |
| can also be stated as the process of validating and verifying that |
| A software program or application or product. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 3) **What is agile methodology?** |
| ANS:- 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.* |
|  |
| Every iteration involves cross functional teams on various |
| Working simultaneously areas like planning, requirements |
| Analysis, design, coding, unit testing, and Acceptance testing. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 4) **what is SRS** |
| ANS:- A software requirements specification (SRS) is a complete |
| Description of the behaviour 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. |
|  |
| This standard describes possible structures, desirable contents, and |
| Qualities of a software requirements specification. |
|  |
| Requirements are categorized in several ways. The following are |
| common categorizations of requirements that relate to technical |
| management: |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **5)** **what is oops** |
| ANS :- 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. |
|  |
| The internal details are hidden. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **6) Write basic concepts of oops** |
| ANS :- Programming is like writing. |
|  |
| If you can write a demonstration, you can make a program. |
|  |
| Programming is not so easy, because a real good program is not. |
| Easily programmed. It needs the programmers’ lots of wisdom, |
| Lots of knowledge about programming and lots of experience. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 7) **What is object** |
| ANS:- Tangible Things Roles = as a car, printer, as contract, sale, .. |
| Incidents = as employee, boss, ... |
| Interactions = as flight, overflow, ... |
| Specifications = as colour, shape, … |
|  |
| An object represents an individual, identifiable item, unit, |
| or entity, either real or abstract, with a well-defined role in the |
| Problem domain. |
|  |
| An "object" is anything to which a concept applies. |
|  |
| This is the basic unit of object oriented programming (OOP). |
|  |
| That is both data and function that operate on data are bundled as |
| a unit called as object. |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 8) **WHAT IS CLASS** |
| ANS :- A class represents an abstraction of the object and abstracts |
| The properties and behaviour of that object. |
|  |
| Class can be considered as the blueprint or definition or a |
| template for an object and describes the properties and |
| Behaviour of that object, but without any actual existence. |
|  |
| An object is a particular instance of a class which has actual |
| existence and there can be many objects (or instances) for a |
| Class. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 9) **What is encapsulation** |
| ANS :- 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. |
|  |
| it is wrapping up of all the data into the single unit is called |
| Encapsulation. |
| E.g. capsule |
|  |
|  |
|  |

|  |
| --- |
| 10) What is inheritance |
| ANS:- Inheritance means that one class inherits the |
| Characteristics of another class. This is also called a |
| “is a” relationship. |
|  |
| One of the most useful aspects of object-oriented |
| Programming is code reusability. As the name. |
| suggests Inheritance is the process of forming a |
| new class from an existing class that is from the |
| existing class called as base class, new class is |
| formed called as derived class |
|  |
|  |
| Types of Inheritance |
| 1) Single-level Inheritance |
| 2) Multi-level Inheritance |
| 3) Multiple-level Inheritance |
| 4) Hierarchical -level Inheritance |
| 5) Hybrid -level Inheritance |
|  |
|  |

|  |
| --- |
| 11) **What is polymorphism** |
| ANS :- 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 ability to change form is known as Polymorphism. |
|  |
|  |
| - poly -> many |
| - morphism -> forms  . |
| E.g. bird and aeroplane. |
| 2 types of polymorphism |
| 1) Compile Time Polymorphism |
| e.g. method overload |
| 2) Run-Time Polymorphism |
| e.g. method overriding |
|  |
|  |
|  |

|  |
| --- |
| 12) DREW USECASE ON ONLINE BOOK SHOPPING |
| ANS :- Shopping book online from amazon |
|  |
|  |
|  |
|  |

***User***

|  |
| --- |
| 13) Draw Use case on online bill payment system |
| (Pay tm) |
| ANS :- Pay online electricity bill payment on pay tm |
|  |

User

|  |
| --- |
| **14 ) Write SDLC phases with basic introduction** |
| **ANS :-** |
| 1 Requirements Collection/ Establish Customer Needs  Gathering |
|  |
| 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 |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| **15) Explain phases of the waterfall model** |
| ANS :- The classical software lifecycle models the software Development |
| As a step-by-step waterfall between the various development phases. |
|  |
| Requirements must be “frozen” To early in the life cycle. |
|  |
| Requirements are validated too late. |
|  |
| Requirements collection |
| 1 Analysis |
| 2 Design |
| 3 Implementation |
| 4 Testing |
| 5 maintenance |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 16) **Write phases of spiral model** |
| **ANS :-** |
|  |
|  |
|  |
| Spiral Model is very widely used in the software industry as it is in |
| synch with the natural development process of any product. |
|  |
|  |
|  |
|  |

|  |
| --- |
| **17) Write agile manifesto principals.** |
| ANS :- In agile there are 4 manifesto . |
|  |
| 1) Individual and Interactions |
| 2) Working software |
| 3) customer collaboration |
| 4) responding to changes |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

|  |
| --- |
| 18) Explain working methodology of agile model |
| and also write pros and cons. |
|  |
| ANS :- 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. |
|  |
| In agile the tasks are divided time boxes (small time Frames) to |
| Deliver specific features for a release. |
|  |
| \*PROS |
| 1 Little or no planning required |
| 2 Easy to manage |
| 3 Gives flexibility to developers |
| 4 Is a very realistic approach to software development |
|  |
| \*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 |
| 4 Must without which it will not work. |

|  |
| --- |
| 19) Draw use case on online shopping product using |
| COD. |
| ANS :- Shopping on Flipkart using cod |
|  |

User

|  |
| --- |
| 20) **Draw use case on online shopping product** |
| **Using payment gateway.** |
|  |

USER