**Software testing assignment**

**Module 1 : fundamentals**

**Q-1. What is SDLC ?**

Ans : software development life cycle is step by step process to develop quality software within time and within budget. 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. Phases of SDLC are given below:

1. Requirment gathering
2. Analisys
3. Designing
4. Implementation
5. Testing
6. Maintainence

**Q-2. What is software testing?**

Ans : Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

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 requirements.

Software testing is a process of executing a program or application with the intent of finding the software bugs.

**Q-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.

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.

**Q-4. What is SRS.**

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

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management:

* Customer Requirements
* unctional Requirements
* Non-Functional Requirements

**Q-5. What is oop.**

Ans : Object-oriented programming

**Q-6. Write Basic Concepts of oops.**

Ans : concepts of oops

* Object :- An instance of a class is an object.
* Class :- Class is a structure in which you can have Member functions & member Variables are there.
* Encapsulation :- Wrapping data into Single unit.
* Inheritance :-To access Property of One class to another Class. There are 5 types of inheritance.

1. Single

2. Multi-level

3. Multiple

4. Hierarchical

5. Hybrid

* Polymorphism :-Same function name but having different Functionalities.

Two types

1. Compile type {overloading}

2. Run type {overriding}

* Abstraction :-Hiding internal details and Showing essential information to user.

**Q-7. What is Object.**

Ans : An instance of a class is an object. 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.

as example; Tangible Things as a car, printer, Roles as employee, boss, ...

**Q-8. What is Class.**

Ans. Class is a structure in which you can have Member functions & member Variables are there. When you define a class, you define a blueprint for an object. A class represents an abstraction of the object and abstracts the properties and behavior of that object. An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class.

**Q-9. What is Encapsulation.**

Ans. Wrapping data into Single unit. 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.

**Q-10. What is Inheritance.**

Ans. To access Property of One class to another Class

**Q-11. What is Polymorphism.**

Ans. Same function name but having different Functionalities. 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.

**Q-12. Draw Use case on Online book shopping.**

Ans.

**Q-13. Draw Use case on Online bill payment** system [ PAYTM]

Ans.

**Q-14. Write SDLC phases with basic introduction.**

Ans : There are six phases in SDLC;

1. Requirements = Establish Customer Needs
2. Analysis = Model And Specify the requirements
3. Design = Model And Specify a Solution
4. Implementation = Construct a Solution In Software
5. Testing = Validate the solution against the requirements
6. Maintenance = Repair defects and adapt the solution

**Q-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.

Since 1950 we are using this model that’s why we call it Traditional model or classical model.

The waterfall is unrealistic for many reasons, especially

* Requirements must be “frozen” to early in the life cycle
* Requirements are validated too late

**Q-16. Write phases of spiral model.**

* Planning = determination of objectives, alternatives and constraints
* Risk Analysis = Analysis of alternatives and identification/ resolution of risks
* Engineering = Development of the "next level" product
* Customer Evaluation = Assessment of the results of engineering

**Q-17. Write agile manifesto principles.**

Ans. 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.

* Individuals and interactions
* Working software
* Customer collaboration
* Responding to change

**Q-18. Explain working methodology of agile model and also write pros and cons.**

Ans. Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability. Each build is incremental in terms of features the final build holds all the features required by the customer.

* Pros :-
* Promotes teamwork and cross training.
* Is a very realistic approach to software development
* Suitable for fixed or changing requirements
* Delivers early partial working solutions.
* Minimal rules, documentation easily employed.
* Cons :-
* More risk of sustainability, maintainability and extensibility.
* Not suitable for handling complex dependencies.
* Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.

**Q-19. Draw use case on online shopping product using COD.**

Ans.

**Q-20. Draw use case on online shopping product using payment gateway.**

Ans. https://drive.google.com/file/d/1\_VWpQcWSAY8jL4gpaA1-MlHHaVHRCr8j/view?usp=drivesdk