1). What Is SDLC?

A. 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.

->. The Methodology Within The SDLC Process Can Vary Across Industries And Organizations, But Standards Such As Iso Represent Processes That Establish A Lifecycle For Software, And Provide A Mode For The Development, Acquisition, And Configuration Of Software Systems.

2). What Is Software Testing?

A. Software Testing Can Be Stated As The Process Of Verifying And Validating Whether A Software Or Application Is Bug-Free, Meets The Technical Requirements As Guided By Its Design And Development, And Meets The User Requirements Effectively And Efficiently By Handling All The Exceptional And Boundary Cases.

->. The Process Of Software Testing Aims Not Only At Finding Faults In The Existing Software But Also At Finding Measures To Improve The Software In Terms Of Efficiency, Accuracy, And Usability.

->. It Mainly Aims At Measuring The Specification, Functionality, And Performance Of A Software Program Or Application.

3). What Is Agile Methodology?

A. Agile Methodology Meaning A Practice That Promotes Continuous Iteration Of Development And Testing Throughout The Software Development Lifecycle Of The Project.

->. In The Agile Model In Software Testing, Both Development And Testing Activities Are Concurrent, Unlike The Waterfall Model.

->. The Agile Methodology Is A Way To Manage A Project By Breaking It Up Into Several Phases. It Involves Constant Collaboration With Stake Holders And Continuous Improvement At Every Stage.

->. Agile Methodology Reorganizes The Software Development Process In Several Key Ways.

4). What Is SRS?

A. 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.

->. Use Cases Are Also Known As Functional Requirements. In Addition To Use Cases, The SRS Also Contains Non-Functional (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).

->. Recommended Approaches For The Specification Of Software Requirements Are Described By IEEE.

->. This Standard Describes Possible Structures, Desirable Contents, And Qualities Of A Software Requirements Specification.

5). What Is Oops?

A.

->. 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.

6). Write Basic Concepts Of Oops?

A.

1. Object

2. Class

3. Encapsulation

4. Inheritance

5. Polymorphism

6. Abstraction

7). What Is Object?

A.

->. Objects Are The Things You Think About First In Designing A Program And They Are Also The Units Of Code That Are Eventually Derived From The Process.

->. In Between, Each Object Is Made In To A Generic Class Of Object, And Even More Generic Classes Are Defined So That Objects Can Share Models And Reuse The Class Definitions In Their Code.

->. Each Object Is An Instance Of A Particular Class Or Subclass With The Class's Own Methods Or Procedures And Data Variables. An Object Is What Actually Runs In The Computer.

8). What Is Class?

A. A Class Representation Abstraction Of The Object And Abstracts The Properties And Behaviour Of That Object.

->. Class Can Be Considered As The Blue Print Ordination 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.

->. In The Case Of A Car Or Laptop, There Will Be A Blue Print 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.

9). What Is Encapsulation?

A. Encapsulation Is The Practice Of Including In An Object Ever Thing 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 To Gether In The Same Object.

->. Encapsulation In Java Is The Process Of Wrapping Up Of Data (Properties) And Behaviour (Methods) Of An Object Into A Single Unit; And The Unit Here Is A Class (Or Interface).

->. Encapsulation Enables Data Hiding, Hiding Irrelevant Information From The Users Of A Class And Exposing Only The Relevant Details Required By The User.

->. 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.

10). What Is Inheritance?

A.

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.

->. This Is A Very Important Concept Of Object-Oriented Programming Since This Feature Helps To Reduce The Code Size.

->. 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.

->. 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.

11). What Is Polymorphism?

A.

Polymorphism Means “Having Many Forms”.

->. It Allows Different Objects To Respond To The Same Message Indifferent Ways, The Response Specific To The Type Of The Object.

->. The Most Important Aspect Of An Object Is Its Behaviour.

->. A Behaviour Is Initiated By Sending A Message To The Object.

->. 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.

->. Poly Refers Too Many. That Is A Single Function Or An Operator Functioning In Many Ways Different Upon The Usage Is Called Polymorphism.

->. Many Ways Different Upon The Usage Is Called Polymorphism.

->. E.G. The Message Display Details Of The Person Class Should Give Different Results When Send To A Student Object.

->. The Ability To Change Form Is Known As Polymorphism

12). Draw Use Case On Online Book Shopping?

A.

13). Draw Use Case On Online Bill Payment System (Paytm)?

A.

14). Write SDLC Phases With Basic Introduction?

A.

1. Requirement Collection: -

->. Requirements Definitions Usually Consist Of Natural Language, Supplemented By Diagrams And Tables.

2. Analysis: -

->. The Analysis Phase Defines The Requirements Of The System, Independent Of How These Requirements Will Be Accomplished.

->. The Deliverable Result At The End Of This Phase Is A Requirement Document.

->. The Deliverable Design Document Is The Architecture.

3. Design: -

->. The Design Team Can Now Expand Upon The Information Established In The Requirement Document. The Requirement Document Must Guide This Decision Process.

->. Analysing The Trade-Offs Of Necessary Complexity Allows For Many Things To Remain Simple Which, In Turn, Will Eventually Lead To A Higher Quality Product.

->. The Architecture Team Also Converts The Typical Scenarios Into A Test Plan.

4. Coding: -

->. In The Implementation Phase, The Team Builds The Components Either From Scratch Or By Composition.

->. Given The Architecture 15 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.

5. Testing: -

->. The Developing Organization Or Team Will Have Some Mechanism To Document And Track Defects And Deficiencies.

->. 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. Quality Is A Distinguishing Attribute Of A System Indicating The Degree Of Excellence.

->. Regression Testing, Internal Testing, Unit Testing, Application Testing, Stress Testing.

6. Maintenance: -

->. 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”! Software Just Like Most Other Products Is Typically Released With A Known Set Of Defects And Deficiencies.

->. The Software Is Released With The Issues Because The Development Organization Decides The Utility And Value Of The Software At A Particular Level Of Quality Outweighs The Impact Of The Known Defects And Deficiencies.

15). Explain Phases Of The Waterfall Model?

A.

1. Requirements And Planning: -

The Requirements And Planning Phase Of Waterfall Project Management Identifies What The Project.

->. This Phase Involves Identifying And Describing The Project’s Risks, Assumptions, Dependencies, Quality Metrics, Costs, And Timeline.

2. Design: -

The Design Phase Solidifies And Documents All Your Decisions. In This Case, You Develop Solutions That Can Solve The Project’s Requirements.

->. Design Covers The Project’s Schedule, Budget, And Objectives, And You Can Think Of Design As A Blueprint Or Road Map To The Complete Project.

3. Implementation: -

The Implementation Phase Executes Your Project Plan And Design To Produce The Desired Product. If Your Company Develops Software, You Will Spend This Phase Coding The Software Functionalities.

->. If You’re Managing A Project At A Construction Company, You Will Construct A House In This Phase.

4. Testing: -

Testing Verifies That The Product Developed In The Implementation Phase Fulfils The Entire Project’s Requirements. If This Is Not The Case, The Project Team Must Review The Project From Phase One To Identify What Went Wrong.

->. The Testing Phase Uses Various Quality Metrics And Customer Satisfaction To Measure The Project’s Success.

5. Maintenance: -

The Maintenance Phase Extends Beyond The Five Stages Of Project Management Into The Project’s Lifetime.

This Phase Involves Making Minor Modifications To Improve The Product Developed During Implementation And Performing Other Routine Maintenance Tasks

16). Write Phases Of Spiral Model?

A.

1. Objectives Determination And Identify Alternative Solutions: -

Requirements Are Gathered From The Customers And The Objectives Are Identified, Elaborated, And Analysed At The Start Of Every Phase.

Then Alternative Solutions Possible For The Phase Are Proposed In This Quadrant.

2. Identify And Resolve Risks: -

During The Second Quadrant, All The Possible Solutions Are Evaluated To Select The Best Possible Solution.

Then The Risks Associated With That Solution Are Identified And The Risks Are Resolved Using The Best Possible Strategy.

At The End Of This Quadrant, The Prototype Is Built For The Best Possible Solution.

3. Develop Next Version Of The Product: -

During The Third Quadrant, The Identified Features Are Developed And Verified Through Testing. At The End Of The Third Quadrant, The Next Version Of The Software Is Available.

4. Review And Plan For The Next Phase: -

In The Fourth Quadrant, The Customers Evaluate The So Far Developed Version Of The Software. In The End, Planning For The Next Phase Is Started.

17). Write Agile Manifesto Principles?

A.

Satisfying Customers Through Early And Continuous Delivery Of Valuable Work.

Breaking Big Work Down Into Smaller Tasks That Can Be Completed Quickly.

Recognizing That The Best Work Emerges From Self-Organized Teams.

Providing Motivated Individuals With The Environment And Support They Need And Trusting Them To Get The Job Done.

Creating Processes That Promote Sustainable Efforts.

Maintaining A Constant Pace For Completed Work.

Welcoming Changing Requirements, Even Late In A Project.

Assembling The Project Team And Business Owners On A Daily Basis Throughout The Project.

Having The Team Reflect At Regular Intervals On How To Become More Effective, Then Tuning And Adjusting Behaviour Accordingly.

Measuring Progress By The Amount Of Completed Work.

Continually Seeking Excellence.

Harnessing Change For A Competitive Advantage.

18). Explain Working Methodology Of Agile Model And Also Write Pros And Cons?

A.

‘Agile Working’ May Sound Like A Classic Case Of Buzzy Business-Speak, But We’re Pleased To Say There’s Real Substance Behind The Name.

->. Agile Is A Very Clearly Defined Work Methodology, Underpinned By An Exceptional Philosophy That Prizes Adaptability And Freedom Over Hierarchy And Dogma.

->. It’s Currently In-Use By Teams At Google, Microsoft, EA And Several Government Agencies Around The World.

->. We’ve Put Together This Article To Explain The Agile Methodology, Unpack Its Components, And Demonstrate How Organisations Large And Small Can Implement It To Their Immense Advantage.

->. We Explain The Technical Language Used To Describe The Aspects Of Agile – Underneath The Surface The Methodology Is Simple And Accessible, Requiring Very Little Technical Knowledge.

19). Draw Use Case On Online Shopping Product Using COD?

A.

Select product add to cart

user

20). Draw Use Case On Online Shopping Product Using Payment Gateway?

A.