Q1. Object oriented technology`s use of \_\_\_\_\_\_\_\_\_\_\_ facilitates reuse of the code and architecture while its \_\_\_\_\_\_\_\_\_\_\_\_\_\_ feature provides systems with stability, as a small change in requirements doesn`t require massive changes in the system.

**1. Inheritance, Encapsulation**

2. Inheritance, Polymorphism

3. Encapsulation, Polymorphism

4. Polymorphism, Abstraction

Q2. Which of the following steps do you think developers should take to create efficient compact applications?

a. Clearly define initial requirements of the system

b. concentrate earlt development efforts on modeling implementation mechanisms

c. Analyze and manage risk throughout the development process

d. Leave all software testing until after system has been implemented

**1. a, c**

2. a, b

3. a, b, d

4. a, b, c

Q3. Which of the following elements combine to form OOAD method

a. Notation

b. Diagram

c. Process

d. View

**1. a, c**

2. a, b

3. a, b, d

4. a, b, c

Q4. Which of the following are aims of UML?

a. To model system using OO concepts

b. To provide a process for software development

c. To support small-scale and large-scale analysis and design

d. To provide an insight into implementation mechanism

1. a, c

2. a, b

3. a, b, d

**4. a, c, d**

Q5. Towards end of the design phase, \_\_\_\_\_\_\_\_\_\_\_ should be allocated to source code components.

1. use cases

2. relationships

3. models

**4. classes**

Q6. What do you think is the first step you should take in designing any project?

1. design a prototype

2. create the test cases

**3. define problem domain and produce problem statement**

4. draw up a plan for entire project

Q7. Which of the following best describes what the problem domain is?

1. kinds of resources available to development team

**2. surroundings in which system operate**

3. set of all functionality required of a system

4. list of technical details needed to implement project

Q8. If you are finding hard to identify the name of class and to write definition for it. What thing you should do?

1. ignore class completely

**2. do more analysis to get a better understanding of what is invaloved in the class**

3. write a definition for the class even if it is not very good

4. make it a friend class of some other main class

Q9. Which of the following statements are true of use cases and use case models?

a. functionality of a use-case has to be complete from start to finish

b. use case provide developers with classes and operations

c. use cases outline functionality of the system

d. use case models can be used to test the system

1. a, b, c

2. a, b, c, d

**3. a, c, d**

4. a, c

Q10. class diagram represents

**1. conceptual design**

2. organization of objects

3. set of actions

4. state machine

Q11. collaboration diagram represents

**1. organization of objects**

2. messages on time scale

3. conceptual design

4. set of actions

Q12. state chart diagram

1. organization of objects

2. conceptual design

3. set of actions

**4. state machine**

Q13. In OOD primary abstraction mechanism is \_\_\_\_\_\_\_\_\_\_

1. function

**2. class**

3. object

4. hierarchy

Q14. requirement analysis \_\_\_\_\_\_\_\_\_\_\_\_\_

1. delivers a system in a series of versions

2. organizes abstraction

**3. builds a bridge between user and developer**

4. uses experimental software to better understand user requirements

Q15. incremental model \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1. delivers a system in a series of versions**

2. works with encapsulation and inheriatance to simplify flow of control

3. builds a bridge between user and developer

4. uses experimental software to better understand user requirements

Q16. prototyping model \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. delivers a system in a series of versions

2. builds a bridge between user and developer

**3. uses experimental software to better understand user requirements**

4. works with encapsulation and inheriatance to simplify flow of control

Q17. software re-engineering actually means reverse enggineers

Correct Answer : F

Q18. re-engineering is a type of software maintainance

Correct Answer : T

Q19. elements of software architecture of a computing systems include

a. software components

b. class diagrams

c. connectors expressing relationships between software components

d. E-R diagram

1. a, b

**2. a, c**

3. a, c, d

4. a, b, c, d

Q20. Project milestones are mainly divided in these two parts

1. DFD and SRS

2. interface design and implementation

3. feasibility study and detailed design

**4. requirements and design**

Q21. Which is not part of testing?

1. white box testing

2. black box testing

**3. inner testing**

4. gorilla testing

Q22. Which is not part of phases of software development

1. high level design

2. low level design

**3. mid level design**

4. replication, delivery, installation

Q23. Which software development model incorporates risk management?

1. water fall model

**2. spiral model**

3. incremental model

4. object model

Q24. largest time is spent on which of the software development phase?

1. testing

**2. enhancement**

3. bug fixing

4. analysis and design

Q25. Simple SDLC contain

**1. requirements, analysis, design, implementation, testing**

2. analysis, design, implementation, testing, deployment

3. analysis, design, implementation, testing, maintainence

4. requirements, analysis, design, implementation, deployment

Q26. DFD is not a

**1. logical model of system**

2. good guide to a system

3. representation of physical stream

4. all of the above

Q27. Productivity metrics

**1. focuses on the output of the development process.**

2. focuses on the characteristics of the software.

3. provide indirect measure.

4. All.

Q28. Which is not a type of maintenance?

1. Adaptive

2. Corrective

3. Perfective

**4. Obsolescence**

Q29. Adaptive Maintenance is

1. To improve the system in some way by changing its basic functionality

**2. The maintenance due to changes in the environment**

3. The correction of undiscovered system errors

4. None of the above

Q30. \_\_\_\_\_\_\_\_\_\_\_ is a method for estimating the software

1. COCOMO

2. Function Point Analysis

3. Use Case Estimation

**4. All of the above**

Q31. Which of the following activities involves choosing a system structure capable of

satisfying the requirement specification?

1. Requirements Analysis

**2. Design**

3. Coding

4. Testing

Q32. Reliability in a software system can be achieved using the following strategies, EXCEPT

1. Fault avoidance

2. Fault tolerance

**3. Fault detection**

4. Fault rectification

Q33. The Software Development Life Cycle covers activities from

1. Feasibility Study to Installation

**2. Requirements Phase to Testing**

3. Requirements Phase to Maintenance

4. Project Initiation to Software Retirement

Q34. Identify the true statements about using a process for software development.

a) Processes usually divide software development into phases

b) Processes provide guidelines for what to do at each phase of development

c) Processes are used o

1. a and c

2. a and b

**3. a, b and d**

4. a, c and d

Q35. Process visibility is enhanced by

1. Defining clear cut phases

2. Producting documents related to each phase

3. Conducting reviews & checks

**4. all of the above**

Q36. Which of the following activities is not considered as "Umbrella Activity"

1. S/W Quality assurance

**2. Software Design**

3. S/W configuration management

4. S/W Project Monitoring & Control

Q37. What is the primary purpose of the first stage of software analysis and design?

1. Determining system deployment

2. Writing code

**3. Capturing requirements**

4. Building GUIs

Q38. Broad design of modules & their relationships is called

1. external design

2. detailed design

**3. architechtural design**

4. process design

Q39. SDLC starts with \_\_\_\_\_\_\_\_\_\_\_ stage

**1. User Requirement and Analysis**

2. Deployment

3. Testing

4. Design

Q40. The following are the steps of SDLC

1. Analysis

2. Design

3. Testing

**4. All of the above**

Q41. The analysis phase takes a \_\_\_\_\_\_ approach to the system, ignoring its inner workings whereas the design phase takes a \_\_\_\_\_ approach, making decisions on how the model will be implemented in code

1. White box & Black box

**2. Black box & White box**

3. Top-Down & Bottom-Up

4. Bottom-Up & Top-Down

Q42. The goal of \_\_\_\_\_\_\_\_ is to obtain a clear understanding of the system and its shortcomings and to determine opportunities for improvement

1. Feasibility study

**2. systems analysis**

3. systems definition

4. systems study

Q43. The last step in System Development Life Cycle is \_\_\_\_\_\_\_\_\_

1. Analysis

2. Implementation

**3. Testing**

4. Maintenance

Q44. The \_\_\_\_\_\_\_\_\_\_ phase of the systems life cycle contains periodic evaluations and updates of the system

1. preliminary investigation

2. Systems analysis

3. Systems implementation

**4. Systems maintenance**

Q45. During the \_\_\_\_\_\_ phase, the application is verified against the requirements

1. Analysis

2. Design

**3. Testing**

4. Implementation

Q46. The type of software maintainence which is done to add new features to the product is called

1. Corrective Maintainence

2. Adaptive Maintainence

3. Regressive Maintainence

4. Perfective Maintainence

Q47. The choice of the Software Development Life Cycle Model to be followed for a project depends on A) Initial Clarity of Requirements B) Size of the Project C) Time Frame of the Project D) Clarity on Technical Issues

1. A, B & C only

2. A, B & D only

**3. A, B, C & D**

4. A & D only

Q48. Because of the cascade from one phase to another, the model of software development process is known as

1. Evolutionary model

2. Formal model

**3. Waterfall model**

4. None of the above

Q49. The Linear Sequential or Classic Life Cycle is also called

**1. Waterfall Model**

2. Incremental Model

3. Spiral model

4. Prototyping Model

Q50. Prototyping in software process may involve \_\_\_\_\_\_.

1. throw - away prototyping

2. evolutionary

**3. Both a and b options**

4. None of these

Q51. Prototype may be used for

1. Risk Reduction

2. Requirements Elicitation

3. User Interface Design

**4. all of the above**

Q52. RAD Model is high speed implementation of

**1. Waterfall Model**

2. Spiral Model

3. Prototyping model

4. Component Assembly model

Q53. Which of the following is not a feature of RAD

1. Well understood, constrained & modularizable requirements

2. Component based construction & use of 4 GL

3. Use of multiple teams each developing separate function

**4. Project has high technical risks**

Q54. Which of the following model may require largest deployment of manpower

1. Incremental Model

2. Waterfall Model

3. Component Assembly Model

**4. RAD Model**

Q55. The majority of the lifetime of a program is spent in the \_\_\_\_\_\_\_\_\_\_ phase

**1. Maintenance**

2. Analysis

3. Design

4. Testing

Q56. Pick up the odd one out of the following process models

1. Component assembly model

2. Prototypiong Model

3. Spiral model

**4. Waterfall Model**

Q57. In Boehm’s spiral model, each loop in the spiral represents \_\_\_\_\_ of the software process

**1. phase**

2. design

3. documentation

4. none of the above

Q58. \_\_\_\_\_\_\_\_ means to build a model that can be modified before the actual system is installed

1. Maintenance

**2. Prototyping**

3. Implementation

4. None of the above

Q59. A requirement may be a description of

1. functionality to be provided

2. constraint on the software

3. external interface

**4. all of the above**

Q60. Which of the following is not true about the context diagram?

1. It does not show details of the funtioning

2. It shows major inputs & outputs of the system

3. It shows the external entities of the system

**4. It shows the datastores of the system**

Q61. Which of the following is seen in the DFD but not in the Context Diagram

1. Data Sources

2. Data Flows

**3. Data Stores**

4. Users

Q62. Data flow cannot take place between

1. a store & a process

2. external entity & process

**3. store & an external entity**

4. peocess & process

Q63. DFD gives idea about flow of \_\_\_\_\_\_\_\_ & flowchart gives idea of the flow of \_\_\_\_\_\_\_\_\_\_\_\_

1. processes, decisions

2. control, data

3. logic, control

**4. data, control**

Q64. Data Models do not consider

1. Attributes of the data object

2. Relationships between data objects

**3. Operations that act on the data**

4. Any of the above

Q65. Which of the following is not a characteristic of a good SRS document?

1. Unambigious

2. Verifiable

**3. Redundant**

4. Consistent

Q66. Notations used to specify the external characteristics, architectural structure, and processing details of a software system include I. Data Flow Diagrams II. HIPO diagrams III. Structure Charts

1. I and II Only

2. III Only

**3. I, II and III**

4. None of the above

Q67. Formal specification language consists of

1. syntax

2. semantics

3. set of relations

**4. all of the above**

Q68. The software architechture is best represented by

1. Context Diagram

2. Flow Chart

**3. Structure Chart**

4. Data Flow Diagram

Q69. Designers should aim to produce strongly \_\_\_\_\_ and weakly \_\_\_\_\_ designs

1. coupled, functional

2. maintainable, cohesive

**3. cohesive, coupled**

4. coupled, cohesive

Q70. Which is the most undesirable form of cohesion from the following options

1. Sequential

**2. Coincidental**

3. Temporal

4. Communicational

Q71. Using \_\_\_\_\_\_\_\_\_\_ a programmer can detail the logic of the program

**1. pseudocode**

2. software

3. context diagram

4. data flow diagram

Q72. Which of the following is not true about a flow chart?

1. It shows the flow of control of a program

2. It is a tool for detailed design

3. Data interchange is not represented

**4. It clearly separates various modules of the software**

Q73. The external interface design process should be \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. developer centered

**2. user centered**

3. administrator centered

4. management centered

Q74. \_\_\_\_\_\_\_ involves modeling a system as a set of interacting functional units.

1. Object oriented decomposition

2. Procedural decomposition

**3. Functional decomposition**

4. None of the above

Q75. Typographical errors and/or incorrect use of the programming language is referred to as

1. logic errors

**2. syntax errors**

3. run time errors

4. A bug

Q76. Testing of software falls after \_\_\_\_\_\_ stage.

1. Designing

2. Implementation

3. Deployment

**4. Coding**

Q77. Changes made to the software to accommodate changes to its environment is called

1. perfective maintainence

2. regressive maintainence

**3. adaptive maintainence**

4. corrective maintainence

Q78. Major changes made to software after long periods is also called software reengineering or

1. perfective maintainence

**2. regressive maintainence**

3. adaptive maintainence

4. corrective maintainence

Q79. Which of the following is not a part of Project Plan?

1. Risk Management Plan

2. Personnel Plan

3. Project Montoring Plan

**4. Software Architechture Planning**

Q80. Function Point Count is dependent on

1. Platform & Technology

2. Team Size

3. H/W & Software Resources

**4. Features & Functionalities**

Q81. In COCOMO terminology a project with mixed level of staff experience & part familiarity with the system being developed is categorized as

1. Organic

**2. Semidetached**

3. Embedded

4. Application

Q82. In COCOMO terminology a project with software being strongly coupled to complex hardware & stringent regulations on operating procedures is categorised as

1. Organic

2. Semidetached

**3. Embedded**

4. Application

Q83. The value of COCOMO cost driver attribute for higher than average Programmer Ability will be

1. Greater than 1

2. Equal to 1

**3. Less than 1**

4. None of these

Q84. \_\_\_\_ and \_\_\_\_ are graphical notations which are used to illustrate the project schedule.

1. Bar chart and DFD

2. ERD and Bar chart

3. Class diagram and activity networks

**4. Bar char and activity networks**

Q85. The total float for an activity is

1. the total duration of the activity

2. the difference between the earliest finish time and earliest start time

**3. the difference between the latest finish time and the earliest finish time**

4. the difference between the latest finish time and the earliest start time

Q86. Most of the project plans should include

1. Risk analysis

2. Project organization

3. Project schedule

**4. All of the above**

Q87. The minimum time required to finish the project can be estimated by considering the \_\_\_\_\_\_ path in the activity graph

1. Shortest

**2. Longest**

3. Average

4. SPT

Q88. According to Putnam the staffing pattern of a software project follows the Rayleigh-Norden curve and peaks during the \_\_\_\_\_\_\_\_\_\_\_\_\_

1. Detailed design

**2. Coding & Unit testing**

3. Integration Testing

4. System Testing

Q89. Chief Programmer Teams are suitable for projects

1. with research orientation

**2. with high modularity**

3. with high creativity

4. None of these

Q90. Arrange the following activities in Risk Assesment in the correct sequence a. Prioritization b. Identification c. Analysis

1. b, a, c

**2. b, c, a**

3. a, b, c

4. c, a, b

Q91. Risk Assesment Table is based on categorization by

1. Risk Components

2. Risk Impact

**3. Both a and b options**

4. None of the above

Q92. The RMMM plan is generally included in the

1. Feasibility Study

**2. Project Plan**

3. SRS Document

4. Project Legacy

Q93. Risks arising out of frequent change requests are best mitigated by

1. User characterization

**2. Strong SCM**

3. Multisource estimations

4. Prescheduling key personnel

Q94. Risk of unrealistic estimates & schedules can be overcome by

1. Using objective methods of estimation rather than judgemental methods

2. Developing a culture of software reuse

3. Performing multisource estimations

**4. all of the above**

Q95. Automated SCM tools help solve problem of

1. Inconsistencies of SCIs

2. concurrent access to SCI

3. instability of development environment

**4. All of these options**

Q96. \_\_\_\_\_\_\_\_\_\_\_ ensures that a set procedure is followed to make any changes to the software

1. Configuration Identification

**2. Configuration Control**

3. Baselining

4. all of the above

Q97. Configuration Management is

1. framework actvity

2. umbrella activity

**3. one time activity**

4. None of the above

Q98. Under SCM the various SCIs are strictly maintained

1. by their respective authors

2. by the appropriate team

**3. in a central project database**

4. all of the above

Q100. As per SEI CMM oganizations which do not have any KPAs present & stable are considered at

**1. Level 1**

2. Level 2

3. Level 3

4. Level 4