Question : 1 After the finalization of SRS, we may like to estimate
A. Size
B. Cost
C. Development time
D. All of the above.

Question : 2 Which one is not a size measure for software
A. LOC
B. Function Count
C. Cyclomatic Complexity
D. Halstead's program length

Question : 3 Function count method was developed by
A. B.Beizer
B. B.Boehm
C. M.halstead
D. Alan Albrecht

Question : 4 Function point analysis (FPA) method decomposes the system into functional units. The total number of functional units are
A. 2
B. 5
C. 4
D. 1

Question : 5 IFPUG stand for
A. Initial function point uniform group
B. International function point uniform group
C. International function point user group
D. Initial function point user group

Question : 6 How many stages are in COCOMO-II?
A. 2
B. 3
C. 4
D. 5

Question : 7 Which one is not a stage of COCOMO-II?
A. Application Composition estimation model
B. Early design estimation model
C. Post architecture estimation model
D. Comprehensive cost estimation model

Question : 8 Function point can be calculated by
A. UFP * CAF
B. UFP * FAC
C. UFP * Cost
D. UFP * Productivity

Question : 9 Putnam resource allocation model is based on
A. Function points
B. Norden/ Rayleigh curve
C. Putnam theory of software management
D. Boehm's observation on manpower utilisation rate

Question : 10 COCOMO was developed initially by
A. B.W.Bohem
B. Gregg Rothermal
C. B.Beizer
D. Rajiv Gupta

Question : 11 A COCOMO model is
A. Common Cost estimation model
B. Constructive cost Estimation model
C. Complete cost estimation model
D. Comprehensive Cost estimation model

Question : 12 Estimation of size for a project is dependent on
A. Cost
B. Schedule
C. Time
D. None of the above

Question : 13 Risk management activities are divided in
A. 3 Categories
B. 2 Categories
C. 5 Categories
D. 10 Categories

Question : 14 Which one is not a risk management activity?
A. Risk assessment
B. Risk control
C. Risk generation
D. None of the above

Question : 15 In function point analysis, number of Complexity adjustment factor are
A. 10
B. 20
C. 14
D. 12

Question : 16 Cost estimation for a project may include
A. Software Cost
B. Hardware Cost
C. Personnel Costs
D. All of the above

Question : 17 In COCOMO model, if project size is typically 2-50 KLOC, then which mode is to be selected?
A. Organic
B. Semidetached
C. Embedded
D. None of the above

Question : 18 COCOMO-II was developed at
A. University of Maryland
B. University of Southern California
C. IBM
D. AT & T Bell labs

Question : 19 Which one is not an infrastructure software?
A. Operating system
B. Database management system
C. Compilers
D. Result management system

Question : 20 The most desirable form of coupling is
A. Control Coupling
B. Data Coupling
C. Common Coupling
D. Content Coupling

Question : 21 The worst type of coupling is
A. Content coupling
B. Common coupling
C. External coupling
D. Data coupling

Question : 22 The most desirable form of cohesion is
A. Logical cohesion
B. Procedural cohesion
C. Functional cohesion
D. Temporal cohesion

Question : 23 The relationship of data elements in a module is called
A. Coupling
B. Cohesion
C. Modularity
D. None of the above

Question : 24 The worst type of cohesion is
A. Temporal cohesion
B. Coincidental cohesion
C. Logical cohesion
D. Sequential cohesion

Question : 25 Which one is not a strategy for design?
A. Bottom up design
B. Top down design
C. Embedded design
D. Hybrid design

Question : 26 Temporal cohesion means
A. Cohesion between temporary variables
B. Cohesion between local variable
C. Cohesion with respect to time
D. Coincidental cohesion

Question : 27 A system that does not interact with external environment is called
A. Closed system
B. Logical system
C. Open system
D. Hierarchal system

Question : 28 Functional cohesion means
A. Operations are part of single functional task and are placed in same procedures
B. Operations are part of single functional task and are placed in multiple procedures
C. Operations are part of multiple tasks
D. None of the above

Question : 29 When two modules refer to the same global data area, they are related as
A. External coupled

B. Data coupled
C. Content coupled
D. Common coupled

Question : 30 The module in which instructions are related through flow of control is
A. Temporal cohesion
B. Logical cohesion
C. Procedural cohesion
D. Functional cohesion

Question : 31 The extent to which different modules are dependent upon each other is called
A. Coupling
B. Cohesion
C. Modularity
D. Stability

Question : 32 Function point can be calculated by
A.UFP * CAF
B.UFP * FAC
C.UFP * Cost
D.UFP * Productivity

Question : 33 Estimation of software development effort for organic software is COCOMO is
A. E = 2.4 (KLOC)1.05 PM
B . E = 3.4 (KLOC)1.06 PM
C. E = 2.0 (KLOC)1.05 PM
D. E = 2.4 (KLOC)1.07 PM

Question : 34 What is Function-Oriented Design primarily focused on?
a) User interface design
b) Organizing software functions
c) Database design
d) Object-oriented programming

Question : 35 In Function-Oriented Design, what is a "module"?
a) A user interface component
b) A high-level software function
c) A database table
d) An object in object-oriented programming

Question : 36 Which of the following is a key principle of Function-Oriented Design?
a) Encapsulation
b) Inheritance

c) Modularity
d) Polymorphism

Question : 37 What is a data flow diagram (DFD) used for in Function-Oriented Design?
a) To represent the flow of control in a program
b) To visualize the data flow between different software modules
c) To show the user interface layout
d) To document the code implementation details

Question : 38 Which of the following is NOT a characteristic of Function-Oriented Design?
a) Emphasis on breaking down the system into functions
b) Focus on data and control flow
c) Extensive use of classes and objects
d) Modularity and reusability

Question : 39 In Function-Oriented Design, what is a "transaction"?
a) A software function that performs a specific task
b) A user's interaction with a system
c) A database record
d) A design pattern

Question : 40 What does "fan-out" refer to in Function-Oriented Design?
a) The number of functions that call a particular function
b) The complexity of a user interface
c) The number of classes in a system
d) The volume of data in a database table

Question : 41 Which phase of software development is Function-Oriented Design typically associated with?
a) Requirements analysis
b) Coding
c) Testing
d) Maintenance

Question : 42 What is the primary approach in top-down design?
a) Start with the smallest components and gradually build up to the entire system.
b) Begin with the overall system and break it down into smaller components.
c) Focus on designing the user interface first.
d) Jump directly into coding without a systematic approach.

Question : 43 In top-down design, which of the following represents the initial focus of design efforts?
a) Detailed algorithms and coding.
b) Subsystems and high-level modules.

c) User interface design.
d) Data structures.

Question : 44 Which design approach is more suitable for complex systems where the overall architecture is not well-defined initially?
a) Top-down design.
b) Bottom-up design.
c) Both approaches are equally suitable.
d) Neither approach is suitable for complex systems.

Question : 45 What is the primary advantage of bottom-up design?
a) It provides a clear and comprehensive overview of the entire system.
b) It is faster and more efficient in terms of development.
c) It allows for early testing and integration of smaller components.
d) It is more suitable for small, simple systems.
Question : 46 Which design approach is often associated with a "divide and conquer" strategy?
a) Top-down design.
b) Bottom-up design.
c) Both top-down and bottom-up design.
d) Neither approach uses a "divide and conquer" strategy.
Question : 47 In bottom-up design, what is typically the starting point for design efforts?
a) The user interface.
b) High-level modules and subsystems.
c) Detailed algorithms and coding.
d) System testing.
Question : 48 Which design approach is often used in agile development methodologies?
a) Top-down design.
b) Bottom-up design.
c) A combination of both approaches.
d) Neither approach is associated with agile methodologies.
Question : 49 In top-down design, what is the term for gradually adding more detail to the design as you move from high-level modules to lower-level modules?
a) Decomposition.
b) Integration.
c) Incremental design.
d) Abstraction.
Question : 50 What is coupling in software design?
a) A measure of how closely related two modules are in terms of their functionality.
b) The process of breaking down a program into smaller functions.
c) The way modules communicate with each other.
d) The level of abstraction used in a module.
Question : 51 High coupling in software design is generally considered:
a) A desirable characteristic that enhances code reusability.
b) A sign of good modular design.

c) A design principle that reduces the likelihood of bugs.
d) Undesirable and can lead to maintenance challenges.

Question : 52 Which type of coupling is generally considered the most desirable in software design?
a) Tight coupling.
b) Loose coupling.
c) Content coupling.
d) Common coupling.

Question : 53 What is "content coupling" in software design?
a) It refers to modules that share a global variable.
b) It describes modules that communicate through a well-defined interface.
c) It represents modules that are closely related in terms of their functionality.
d) It is a measure of the similarity of data structures in different modules.

Question : 54 In software design, what is meant by "data coupling"?
a) Modules communicate by passing data rather than calling each other's functions.
b) Modules use a common global variable to store data.
c) Modules share similar data structures.
d) Modules communicate through a well-defined interface.

Question : 55 Which of the following is an example of "stamp coupling"?
a) Module A uses a function from Module B, but the two modules have no other connections.
b) Module A and Module B communicate through a global variable.
c) Module A and Module B share the same data structure.
d) Module A calls functions in Module B, and Module B calls functions in Module C.

Question : 56 Why is minimizing coupling important in software design?
a) It simplifies the code and makes it more readable.
b) It reduces the likelihood of bugs and makes the code easier to maintain.
c) It speeds up program execution by reducing function calls.
d) It allows for better distribution of code across multiple files.

Question : 57 What is cohesion in software design?
a) A measure of how closely related two modules are in terms of their functionality.
b) The process of breaking down a program into smaller functions.
c) The way modules communicate with each other.
d) A measure of how closely the elements within a module are related in terms of their functionality.

Question : 58 High cohesion in software design is generally considered:
a) A sign of good modular design.
b) A measure of how loosely connected modules are.
c) A design principle that promotes extensive coupling.
d) Undesirable and can lead to maintenance challenges.

Question : 59 Which type of cohesion is generally considered the most desirable in software design?
a) Functional cohesion.
b) Coincidental cohesion.
c) Temporal cohesion.

d) Procedural cohesion.

Question : 60 What is "functional cohesion" in software design?
a) It refers to modules that perform a single, well-defined task.
b) It describes modules that have no clear purpose or organization.
c) It represents modules that contain unrelated functions.
d) It is a measure of how often a module is used in the program.

Question : 61 In software design, what is meant by "temporal cohesion"?
a) Modules perform tasks that are related in time, such as initialization and cleanup.
b) Modules are organized based on the order they were written.
c) Modules are organized based on their file location.
d) Modules are cohesive due to their common use of global variables.

Question : 62 Which of the following is an example of "sequential cohesion"?
a) Module A contains functions related to file I/O, and Module B contains functions related to database operations.
b) Module A contains functions related to data processing, and Module B contains functions related to user interface.
c) Module A and Module B contain functions that are executed one after the other.
d) Module A contains functions related to mathematical calculations, and Module B contains functions for generating reports.

Question : 63 Why is maximizing cohesion important in software design?
a) It simplifies the code and makes it more readable.
b) It reduces the likelihood of bugs and makes the code easier to maintain.
c) It increases the amount of code that can be reused.
d) It allows for better distribution of code across multiple files.

Question : 64 What is the primary purpose of using design notation in software engineering?
a) To create colorful and visually appealing diagrams.
b) To provide a standardized and precise way of representing design concepts.
c) To write code directly without the need for documentation.
d) To simplify the design process by using natural language.

Question : 65 Which design notation is commonly used for representing the structure of a system's classes and their relationships in object-oriented design?
a) UML (Unified Modeling Language)
b) ERD (Entity-Relationship Diagram)
c) DFD (Data Flow Diagram)
d) IDEF0 (Integrated Definition for Function Modeling)

Question : 66 In UML (Unified Modeling Language), what does a solid line with a filled arrowhead indicate in a class diagram?
a) Inheritance or generalization relationship.
b) Composition relationship.
c) Aggregation relationship.
d) Dependency relationship.

Question : 67 Which type of diagram in UML is used to represent the dynamic behavior of a system by showing the interactions between objects or components?
a) Class diagram

b) Sequence diagram
c) Use case diagram
d) State diagram

Question : 68 What does a rectangle in an Entity-Relationship Diagram (ERD) represent?
a) A relationship between entities.
b) An attribute of an entity.
c) An entity or table.
d) A link to external data sources.

Question : 69 Which design notation is commonly used to model data flow within a system and how data is transformed from one process to another?
a) UML (Unified Modeling Language)
b) ERD (Entity-Relationship Diagram)
c) DFD (Data Flow Diagram)
d) IDEF0 (Integrated Definition for Function Modeling)

Question : 70 In a Data Flow Diagram (DFD), what does a rounded rectangle symbolize?
a) A data store or repository.
b) A process or function.
c) A data flow or data transfer.
d) An external entity.

Question : 71 What is the primary purpose of design specifications in software engineering?
a) To describe the physical appearance of the user interface.
b) To document the complete source code for the software.
c) To provide detailed information about how the software should be designed and built.
d) To list the names of the project team members.

Question : 72 What is the primary goal of Structured Design Methodology in software engineering?
a) To create a visually appealing user interface.
b) To develop detailed code.
c) To provide a systematic approach to designing and building software.
d) To ensure rigorous testing of the software.

Question : 73 In Structured Design, what is the purpose of a Data Flow Diagram (DFD)?
a) To represent the flow of control in a program.
b) To visualize the user interface design.
c) To depict the flow of data and information between different components of a system.
d) To document the code implementation details.

Question : 74 Which of the following is a key concept in Structured Design Methodology?
a) Object-oriented programming.
b) Flowcharts.
c) Modularity and decomposition.
d) Database normalization.

Question : 75 What is the primary purpose of a Structure Chart in Structured Design?
a) To illustrate the user interface layout.
b) To provide a detailed algorithm for a specific function.
c) To represent the hierarchy and organization of software modules.

d) To show the data entities and their relationships.

Question : 76 In Structured Design, what is "top-down design"?

a) Starting with the smallest components and gradually building up to the entire system.
b) Beginning with the overall system and breaking it down into smaller components.
c) Creating a user interface first before any other design work.
d) Skipping the design phase and jumping directly into coding.

Question : 77 Which design concept in Structured Design encourages breaking down a system into smaller, manageable modules?

a) Abstraction.
b) Encapsulation.
c) Modularity.
d) Inheritance.

Question : 78 What does "cohesion" refer to in Structured Design?

a) The degree to which modules are closely related in terms of their functionality.
b) The way modules communicate with each other.
c) The design of the user interface.
d) The level of abstraction used in a module.

Question : 79 What is "coupling" in Structured Design?

a) The degree of interdependence between modules.
b) The organization of data within a module.
c) The flow of control within a program.
d) The way modules are connected to external databases.

Question : 80 In Structured Design, what is the purpose of a Decision Table?

a) To represent the flow of data within a system.
b) To illustrate the hierarchy of software modules.
c) To document user requirements.
d) To provide a structured way to represent different conditions and their outcomes.

Question : 81 Which of the following is a common method for representing control flow in Structured Design?

a) Data Flow Diagram (DFD).
b) Entity-Relationship Diagram (ERD).
c) Flowchart.
d) Object-Role Modeling (ORM).

Question :82 What is the primary focus of the "module specification" phase in Structured Design?

a) Writing code for individual modules.
b) Documenting how different modules interact with each other.
c) Describing the detailed design of each module, including input, processing, and output.
d) Preparing user manuals.

Question : 83 In Structured Design, what does "structured walk-through" mean?

a) Walking through the physical location of different team members.
b) A review process to ensure that the design conforms to the standards and is logically sound.
c) The process of running a software program to identify bugs.
d) A user testing phase.

Question : 84 What is the primary purpose of "pseudo-code" in Structured Design?
a) To create a visual representation of the user interface.
b) To document the complete source code for the software.
c) To provide a high-level, human-readable representation of the program's logic.
d) To create test cases for software testing.
Question : 85 Which design concept in Structured Design promotes the use of well-defined interfaces between modules?
a) Encapsulation.
b) Abstraction.
c) Modularity.
d) Inheritance.
Question : 86 Which design method is often associated with "bottom-up design" in Structured Design?
a) Data Flow Diagram (DFD).
b) Entity-Relationship Diagram (ERD).
c) Structured Analysis.
d) Object-Oriented Programming (OOP).
Question : 87 What is the primary objective of project planning?
a) To ensure the project is completed ahead of schedule.
b) To define the project scope and objectives.
c) To minimize project costs at all costs.
d) To secure funding for the project.
Question : 88 What is the purpose of defining project objectives during the planning phase?
a) To outline the project's timeline and milestones.
b) To identify potential risks and issues.
c) To establish clear, measurable goals for the project.
d) To allocate resources to specific tasks.
Question : 89 Which of the following is NOT a common project planning objective?
a) To manage project stakeholders' expectations.
b) To maintain a high level of team morale throughout the project.
c) To deliver the project within budget.
d) To achieve the project's intended benefits.
Question : 90 When defining project objectives, what does the SMART acronym stand for?
a) Specific, Measurable, Achievable, Realistic, Timely.
b) Simple, Meaningful, Actionable, Relevant, Time-bound.
c) Specific, Meaningful, Achievable, Resourceful, Trackable.
d) Systematic, Measurable, Adaptive, Realistic, Targeted.
Question : 91 Which project planning objective focuses on ensuring that the project's goals align with the organization's strategic objectives?
a) Cost management.
b) Risk management.
c) Scope management.
d) Alignment with organizational strategy.
Question : 92 What is the role of a project manager in relation to project planning objectives?

a) The project manager is responsible for defining the project objectives.
b) The project manager ensures that project objectives are met by overseeing the project's execution.
c) The project manager only communicates with stakeholders but does not influence project objectives.
d) The project manager has no involvement in setting project objectives.

Question : 93 Which project planning objective is concerned with identifying and managing potential issues and challenges that may impact the project's success?
a) Time management.
b) Quality management.
c) Risk management.
d) Cost management.

Question : 94 Which project planning objective deals with allocating resources efficiently to complete project tasks?
a) Scope management.
b) Resource management.
c) Stakeholder management.
d) Communication management.

Question : 95 Which of the following is an example of a well-defined project objective?
a) "Complete the project as soon as possible."
b) "Reduce customer support response times by 25% within six months."
c) "Spend less money on the project than last year."
d) "Increase team productivity."

Question : 96 What is the significance of having clear and well-defined project objectives?
a) They make it easier to allocate blame when a project fails.
b) They provide a framework for monitoring and measuring project progress.
c) They ensure that the project is completed faster.
d) They eliminate the need for project planning altogether.

Question : 97 What is the primary purpose of software metrics in software engineering?
a) To create more complex and feature-rich software products.
b) To monitor the performance of hardware components in a computer system.
c) To quantify and measure various aspects of the software development process and the resulting software products.
d) To enforce strict coding standards.

Question : 98 Which of the following is NOT a common category of software metrics?
a) Process metrics.
b) Product metrics.
c) System metrics.
d) Project metrics.

Question : 99 Which type of software metric is concerned with measuring the number of defects or issues found in the software?
a) Product metrics.
b) Process metrics.
c) Project metrics.

d) System metrics.

Question : 100 What is the purpose of using "code complexity metrics" in software development?
a) To measure the complexity of the software development process.
b) To identify bugs in the code.
c) To assess the complexity of the source code, which may affect maintainability and quality.
d) To calculate the project budget.

Question : 101 Which of the following is a commonly used software metric for code size in lines of code (LOC)?
a) McCabe's Cyclomatic Complexity.
b) Halstead's Software Science.
c) Function Points.
d) LOC.

Question : 102 What does "defect density" refer to in software metrics?
a) The number of defects in a software project relative to the project's size or lines of code.
b) The number of developers assigned to a project.
c) The complexity of the software.
d) The percentage of project milestones completed.

Question : 103 Which type of software metric measures the amount of time it takes to complete a specific task or project?
a) Product metrics.
b) Process metrics.
c) System metrics.
d) Project metrics.

Question : 104 What is the primary goal of "function points" as a software metric?
a) To measure the complexity of the source code.
b) To estimate the effort required to develop a software system.
c) To identify security vulnerabilities.
d) To assess the quality of the user interface.

Question : 105 In software development, what does "COCOMO" stand for?
a) Comprehensive Code Modeling.
b) Constructive Cost Model.
c) Code Complexity Measurement.
d) Complexity of Compiled Objects.

Question : 106 Which software metric is used to evaluate the maintainability and readability of source code?
a) Halstead's Software Science.
b) McCabe's Cyclomatic Complexity.
c) Code coverage.
d) Code readability index.

Question : 107 In the context of software metrics, what is "software quality"?
a) The number of lines of code in the software.
b) The extent to which software meets its requirements and user expectations.
c) The cost of the software development project.

d) The level of complexity in the source code.

Question : 108 What is the primary objective of using "effort estimation metrics" in software development?

a) To determine the complexity of the software architecture.
b) To calculate the time and resources required to complete a software project.
c) To identify security vulnerabilities.
d) To assess the user experience design.

Answers:

1. D
2. C
3. D
4. B
5. C
6. B
7. D
8. A
9. B
10. A
11. B
12. D
13. B
14. C
15. C
16. D
17. A
18. B
19. D
20. B
21. A
22. C
23. B
24. B
25. C
26. C
27. A
28. A
29. D
30. A
31. A
32. A
33. C

34. Organizing software functions
35. A high-level software function
36. Modularity
37. To visualize the data flow between different software modules
38. Extensive use of classes and objects
39. A software function that performs a specific task
40. The number of functions that call a particular function
41. Requirements analysis
42. b) Begin with the overall system and break it down into smaller components.
43. b) Subsystems and high-level modules.
44. b) Bottom-up design.
45. c) It allows for early testing and integration of smaller components.
46. a) Top-down design.
47. c) Detailed algorithms and coding.
48. c) A combination of both approaches.
49. c) Incremental design.
50. a) A measure of how closely related two modules are in terms of their functionality.
51. d) Undesirable and can lead to maintenance challenges.
52. b) Loose coupling.
53. c) It represents modules that are closely related in terms of their functionality.
54. a) Modules communicate by passing data rather than calling each other's functions.
55. a) Module A uses a function from Module B, but the two modules have no other connections.
56. b) It reduces the likelihood of bugs and makes the code easier to maintain.
57. d) A measure of how closely the elements within a module are related in terms of their functionality.
58. a) A sign of good modular design.
59. a) Functional cohesion.
60. a) It refers to modules that perform a single, well-defined task.
61. a) Modules perform tasks that are related in time, such as initialization and cleanup.
62. c) Module A and Module B contain functions that are executed one after the other.
63. b) It reduces the likelihood of bugs and makes the code easier to maintain.
64. b) To provide a standardized and precise way of representing design concepts.
65. a) UML (Unified Modeling Language)
66. a) Inheritance or generalization relationship.
67. b) Sequence diagram
68. c) An entity or table.
69. c) DFD (Data Flow Diagram)
70. a) A data store or repository.
71. c) To provide detailed information about how the software should be designed and built.
72. c) To provide a systematic approach to designing and building software.
73. c) To depict the flow of data and information between different components of a system.
74. c) Modularity and decomposition.
75. c) To represent the hierarchy and organization of software modules.

76. b) Beginning with the overall system and breaking it down into smaller components.
77. c) Modularity.
78. a) The degree to which modules are closely related in terms of their functionality.
79. a) The degree of interdependence between modules.
80. d) To provide a structured way to represent different conditions and their outcomes.
81. c) Flowchart.
82. c) Describing the detailed design of each module, including input, processing, and output.
83. b) A review process to ensure that the design conforms to the standards and is logically sound.
84. c) To provide a high-level, human-readable representation of the program's logic.
85. a) Encapsulation.
86. c) Structured Analysis.
87. b) To define the project scope and objectives.
88. c) To establish clear, measurable goals for the project.
89. b) To maintain a high level of team morale throughout the project.
90. a) Specific, Measurable, Achievable, Realistic, Timely.
91. d) Alignment with organizational strategy.
92. b) The project manager ensures that project objectives are met by overseeing the project's execution.
93. c) Risk management.
94. b) Resource management.
95. b) "Reduce customer support response times by 25% within six months."
96. b) They provide a framework for monitoring and measuring project progress.
97. c) To quantify and measure various aspects of the software development process and the resulting software products.
98. c) System metrics.
99. a) Product metrics.
100. c) To assess the complexity of the source code, which may affect maintainability and quality.
101. d) LOC.
102. a) The number of defects in a software project relative to the project's size or lines of code.
103. d) Project metrics.
104. b) To estimate the effort required to develop a software system.
105. b) Constructive Cost Model.
106. d) Code readability index.
107. b) The extent to which software meets its requirements and user expectations.
108. b) To calculate the time and resources required to complete a software project.