SEN 201 CBT QUESTIONS FOR PREPARATION (Note that these aren't actual questions, this is just for you to prepare yourself after reading your PDF)

1. Software engineering is mainly concerned with

A. Data communication

B. Building computer hardware

C. Computer networking

D. Systematic development and maintenance of software

2. Which of the following best defines software

A. Internet services

B. Programs and associated documentation

C. Computer hardware components

D. Only computer programs

3. An attribute of software that makes it easy to modify when changes occur is known as

A. Portability

B. Reliability

C. Maintainability

D. Efficiency

4. Software differs from hardware because software

A. Is manufactured in factories

B. Does not suffer physical deterioration

C. Can physically wear out

D. Cannot be reused

5. The main objective of software engineering is to

A. Avoid testing

B. Write complex programs

C. Produce quality software within time and budget

D. Eliminate documentation

6. A software process can best be described as

A. A database system

B. A programming language

C. A hardware specification

D. A sequence of activities involved in software development

7. Which of the following is a generic software process activity

A. Hardware installation

B. Internet browsing

C. Coding only

D. Requirements engineering

8. The waterfall model is characterized by

A. Continuous prototyping

B. Parallel development phases

C. Sequential phases with little overlap

D. Random development stages

9. One major limitation of the waterfall model is that it

A. Encourages early delivery

B. Eliminates documentation

C. Is too flexible

D. Does not handle changing requirements well

10. Incremental development involves

A. Avoiding testing

B. Delivering the system in small usable parts

C. Delivering software only once

D. Completing all design before coding

11. Reuse oriented software engineering emphasizes

A. Ignoring design principles

B. Writing new code for every system

C. Using existing components and systems

D. Avoiding system integration

12. Requirements engineering focuses mainly on

A. Managing hardware resources

B. Writing source code

C. Understanding what users need

D. Testing completed systems

13. Functional requirements specify

A. What the system should do

B. Programming tools

C. System development cost

D. System quality attributes

14. Non functional requirements are concerned with

A. Algorithms

B. Database tables

C. System services only

D. Quality attributes such as performance and security

15. User requirements are written in a simple form mainly for

A. Network administrators

B. End users and stakeholders

C. Software developers only

D. Computer hardware engineers

16. System requirements are more detailed because they

A. Avoid technical details

B. Ignore constraints

C. Serve as a basis for system design

D. Are written in natural language

17. Requirements validation ensures that

A. Documentation is finished

B. Code runs without error

C. Requirements reflect what the customer wants

D. Testing is complete

18. Requirements management is important because

A. Changes to requirements must be controlled

B. Coding can be avoided

C. Testing is unnecessary

D. Requirements never change

19. System modeling is mainly used to

A. Write source code

B. Replace system testing

C. Reduce project cost

D. Visualize and understand system structure and behavior

20. UML stands for

A. Universal Modeling Logic

B. Unified Modeling Language

C. Universal Machine Language

D. User Modeling Logic

21. Context models show

A. System boundaries and external interactions

B. Memory allocation

C. Internal data structures

D. System algorithms

22. Interaction models focus on

A. Database normalization

B. Hardware connections

C. Communication between users and the system

D. Network topology

23. Structural models describe

A. User interface layout

B. Organization of system components

C. System performance

D. Dynamic system behavior

24. Behavioral models are used to represent

A. File storage format

B. Coding syntax

C. Static structure

D. Dynamic behavior over time

25. Software architecture refers to

A. Hardware layout

B. Programming speed

C. Overall structure and organization of a system

D. Program documentation

26. An architectural pattern provides

A. Testing tools

B. Database schemas

C. Source code templates

D. A reusable solution to common design problems

27. Client server architecture involves

A. Servers requesting services from clients

B. Standalone systems only

C. Clients requesting services from servers

D. Independent systems without communication

28. A key advantage of layered architecture is

A. Lack of structure

B. Poor performance

C. Ease of maintenance and modification

D. Increased complexity

29. Software design focuses on

A. Running test cases

B. Installing software

C. Marketing software products

D. Deciding how the system will be built

30. Object oriented design is based on the concept of

A. Network protocols

B. Objects and classes

C. Machine instructions

D. Functions only

31. Database design is concerned with

A. User interface layout

B. Software testing

C. Code compilation

D. Organizing data efficiently

32. Interface design focuses on

A. Hardware performance

B. Database queries

C. Interaction between users and the system

D. System memory usage

33. Debugging is the process of

A. Designing systems

B. Writing documentation

C. Installing software

D. Finding and fixing program errors

34. Software testing is carried out mainly to

A. Replace debugging

B. Detect defects in the software

C. Speed up coding

D. Prove software is error free

35. Verification checks whether

A. Users are satisfied

B. The software is built correctly

C. The system meets user needs

D. The project is profitable

36. Validation checks whether

A. Documentation exists

B. Code compiles successfully

C. The right system is built

D. The system is cheap

37. Unit testing focuses on

A. Individual program components

B. User acceptance

C. Network performance

D. The entire system

38. Integration testing is used to test

A. Hardware devices

B. User manuals

C. Combined system components

D. Single modules

39. System testing involves

A. Code inspection only

B. Database design

C. Testing individual functions

D. Testing the complete integrated system

40. Acceptance testing is usually performed by

A. Network administrators

B. Hardware engineers

C. Software developers

D. End users or clients

41. Test driven development emphasizes

A. Writing tests before code

B. Manual testing only

C. Writing code before tests

D. Avoiding testing

42. Software evolution refers to

A. Programming language updates

B. Hardware upgrades

C. Initial software development

D. Changes made after deployment

43. Software maintenance includes

A. Writing source code only

B. Modifying and improving software after delivery

C. Fixing errors only

D. System installation

44. Corrective maintenance deals with

A. Improving performance

B. Adding new features

C. Fixing discovered faults

D. Adapting to new environments

45. Adaptive maintenance focuses on

A. Bug fixing

B. Removing system features

C. Code refactoring

D. Adjusting software to new environments

46. Software dependability includes

A. Programming speed

B. User friendliness only

C. Cost effectiveness

D. Reliability, safety, and security

47. Software reliability refers to the

A. User interface quality

B. Probability of failure free operation

C. Size of the program

D. Speed of execution

48. Software security is mainly concerned with

A. Enhancing graphics

B. Reducing development time

C. Improving performance

D. Protecting systems from threats and attacks

49. A software project risk is best described as

A. A system feature

B. A completed task

C. A potential problem that may affect the project

D. A guaranteed problem

50. Software project management mainly involves

A. Planning, scheduling, and controlling software projects

B. Writing source code

C. Designing databases

D. Running test cases

51. Project planning in software engineering is mainly concerned with

A. Debugging programs

B. Writing source code

C. Estimating time, cost, and required resources

D. Designing interfaces

52. A software project schedule shows

A. Hardware layout

B. Timeline of project activities

C. Database structure

D. Program logic

53. A software risk can best be described as

A. A system requirement

B. A problem that has already occurred

C. A completed task

D. A potential problem that may affect project success

54. Risk identification involves

A. Writing code early

B. Testing completed software

C. Ignoring possible problems

D. Listing potential risks in a project

55. Risk analysis is concerned with

A. Assessing likelihood and impact of risks

B. Eliminating all risks

C. Hiring developers

D. Writing documentation

56. Risk planning focuses on

A. Coding system modules

B. Writing test cases

C. Developing strategies to handle risks

D. Database normalization

57. Risk monitoring involves

A. Tracking identified risks throughout the project

B. Writing new requirements

C. Installing software

D. Ignoring resolved risks

58. People management in software projects is important because

A. Documentation replaces teamwork

B. Software development is done by teams of people

C. Hardware is more important than people

D. Software is fully automated

59. A major challenge in software project management is

A. Writing short programs

B. Choosing a programming font

C. Managing changing requirements

D. Hardware availability

60. Software evolution becomes necessary mainly because

A. Software automatically expires

B. Hardware stops working

C. Users stop using the system

D. Business and user needs change over time

61. Perfective maintenance involves

A. Adapting software to new platforms

B. Fixing system faults

C. Improving system performance or features

D. Removing unused code

62. Preventive maintenance focuses on

A. Adding new requirements

B. Fixing known bugs

C. System installation

D. Reducing future system problems

63. Refactoring refers to

A. Removing system documentation

B. Writing new requirements

C. Changing software behavior

D. Improving code structure without altering functionality

64. Dependable software is expected to

A. Have complex code

B. Run only on specific hardware

C. Be cheap to produce

D. Deliver services that can be trusted

65. Software safety is mainly concerned with

A. Enhancing graphics

B. Preventing failures that may cause harm

C. Improving execution speed

D. Reducing memory usage

66. A system vulnerability refers to

A. A testing method

B. A system feature

C. A weakness that can be exploited

D. A programming language

67. Distributed systems are systems in which

A. Software runs offline

B. Only hardware is distributed

C. All components run on one computer

D. Components are located on different networked computers

68. In a client server system, the client typically

A. Provides services

B. Stores databases only

C. Requests services

D. Controls the network

69. In client server architecture, the server is responsible for

A. Writing program code

B. Managing user interfaces

C. Providing services to clients

D. Requesting services

70. Multitier architecture separates a system into

A. Single processing units

B. Independent random modules

C. Hardware layers only

D. Presentation, application, and data layers

71. Peer to peer systems differ from client server systems because

A. They rely on centralized control

B. All nodes can act as both clients and servers

C. Only servers exist

D. They have no communication

72. Service oriented architecture is based on

A. Hardware components

B. Standalone programs

C. Independent services communicating over a network

D. Programming syntax

73. Web services allow systems to

A. Communicate using standard web protocols

B. Replace databases

C. Run without internet

D. Avoid system integration

74. Remote procedure call allows a program to

A. Avoid network communication

B. Replace user interfaces

C. Execute only local functions

D. Invoke procedures on remote systems

75. Message based communication differs from RPC because it

A. Requires no network

B. Eliminates services

C. Uses explicit message exchange

D. Is synchronous only

76. Software validation is important because it ensures that

A. Documentation is complete

B. The delivered system meets user needs

C. Code compiles correctly

D. Testing is skipped

77. Debugging differs from testing because debugging focuses on

A. Preventing failures

B. Writing requirements

C. Detecting faults only

D. Locating and fixing identified faults

78. A test case is best described as

A. A design diagram

B. A section of code

C. A set of inputs and expected outputs

D. A project plan

79. Black box testing focuses on

A. Compiler errors

B. Database schemas

C. Internal code structure

D. System functionality without considering internal design

80. White box testing examines

A. External behavior only

B. User documentation

C. Internal logic and code structure

D. Network performance

81. Regression testing is carried out to

A. Replace system testing

B. Eliminate debugging

C. Test new hardware

D. Ensure changes have not introduced new faults

82. Software documentation is important because it

A. Replaces code

B. Eliminates maintenance

C. Helps users and developers understand the system

D. Increases program size

83. User documentation is primarily written for

A. End users

B. Network administrators

C. Database engineers

D. System designers

84. Technical documentation is mainly intended for

A. Hardware vendors

B. Customers only

C. Developers and maintainers

D. End users

85. Configuration management deals with

A. Writing test cases

B. Hardware installation

C. Project scheduling

D. Controlling changes to software artifacts

86. Version control systems are used to

A. Design interfaces

B. Compile programs

C. Track changes to source code

D. Test software

87. A baseline in configuration management refers to

A. Final product only

B. Approved version of a software artifact

C. Hardware specification

D. Initial project idea

88. Software quality assurance focuses on

A. Ensuring software meets defined standards

B. Reducing team size

C. Writing code faster

D. Eliminating documentation

89. Reviews and inspections are mainly used to

A. Replace testing

B. Execute software

C. Increase development time

D. Detect defects early

90. A feasibility study is conducted to

A. Install hardware

B. Test completed software

C. Assess whether a project is viable

D. Write program code

91. Economic feasibility focuses on

A. User satisfaction

B. System security

C. Cost and benefit analysis

D. Technical performance

92. Technical feasibility examines whether

A. Required technology is available

B. The system is cheap

C. Documentation exists

D. Users like the system

93. Operational feasibility considers

A. Hardware cost

B. Network speed

C. Code structure

D. Whether users can operate the system effectively

94. Software reuse improves productivity because it

A. Avoids testing

B. Increases project risk

C. Eliminates documentation

D. Reduces development effort

95. A prototype is mainly used to

A. Replace the final system

B. Eliminate coding

C. Clarify requirements and design decisions

D. Reduce documentation

96. Throwaway prototyping means the prototype is

A. Deployed to users

B. Used as final product

C. Continuously improved

D. Discarded after requirements are understood

97. Evolutionary prototyping involves

A. Eliminating design

B. Avoiding user feedback

C. Gradually refining the prototype into the final system

D. Discarding prototypes

98. A major benefit of prototyping is that it

A. Avoids testing

B. Increases project cost

C. Helps users clarify their requirements
D. Eliminates maintenance

99. Overall, software engineering aims to produce software that is
A. Large in size
B. Written in one programming language
C. Complex and expensive
D. Reliable, maintainable, and meets user needs

100. Software project management ensures that software is developed
A. Without documentation
B. With focus on hardware only
C. According to plan, time, and budget
D. Without user involvement

SEN 201 CBT ANSWERS (1–100)

1-D, 2-B, 3-C, 4-B, 5-C, 6-D, 7-D, 8-C, 9-D, 10-B,
11-C, 12-C, 13-A, 14-D, 15-B, 16-C, 17-C, 18-A, 19-D, 20-B,
21-A, 22-C, 23-B, 24-D, 25-C, 26-D, 27-C, 28-C, 29-D, 30-B,
31-D, 32-C, 33-D, 34-B, 35-B, 36-C, 37-A, 38-C, 39-D, 40-D,
41-A, 42-D, 43-B, 44-C, 45-D, 46-D, 47-B, 48-D, 49-C, 50-A,
51-C, 52-B, 53-D, 54-D, 55-A, 56-C, 57-A, 58-B, 59-C, 60-D,
61-C, 62-D, 63-D, 64-D, 65-B, 66-C, 67-D, 68-C, 69-C, 70-D,
71-B, 72-C, 73-A, 74-D, 75-C, 76-B, 77-D, 78-C, 79-D, 80-C,
81-D, 82-C, 83-A, 84-C, 85-D, 86-C, 87-B, 88-A, 89-D, 90-C,
91-C, 92-A, 93-D, 94-D, 95-C, 96-D, 97-C, 98-C, 99-D, 100-C