BCA _II

Paper Code-BCA-203 Software Engineering

QUESTION BANK

1. What is software engineering and why is it important?
2. What are the phases of the software development life cycle (SDLC)?
3. What is the difference between software testing and software debugging?
4. What is version control and why is it important in software engineering?
5. What is software documentation and why is it important?
6. What is the difference between a functional requirement and a non-functional requirement?
7. What is the difference between a software application n a and a software s system?
8. What is a problem domain in software engineering?
9. Why is it important to understand the problem domain before designing and Implementing software?
10. What are some common techniques used to gather information about a problem Domain?
11. What is the difference between the problem domain and the solution domain?
12. How can a software developer ensure that they are addressing the correct problem domain?
13. What are some challenges that can arise when working with a complex problem domain?
14. What is domain modeling and how can it be used to represent the problem domain?
15. How can domain experts be involved in the software development process to ensure a
16. How can ensure a thorough understanding of the problem domain? thorough understanding of the problem domain?
17. What is domain-driven design (DDD) and how does it help address problem domain complexity?
18. How can changes in the problem domain be accommodated during the software development process?
19. What are some common software engineering challenges that developers face?

20. How can poor software design impact a project?
21. What are some common software security issues and how can they be prevented?
22. What are some strategies for managing technical debt in software development?
23. How can software engineers ensure that their code is compatible with different Platforms and devices?
24. How can software engineers ensure that their code is performant and efficient?
25. What is a software process, and why is it important?
26. What are the main stages of the waterfall model of software development?
27. What is the agile software development process, and how does it differ from the waterfall model?
28. What is DevOps, and how does it relate to software processes?
29. What are some common software process models, and when might each be most appropriate to use?
30. How can software processes be tailored to fit the needs of a specific project or team?
31. What are some challenges that can arise when implementing software processes, and how can they be addressed?
32. How can software processes be measured and evaluated for effectiveness?
33. What is the agile development model, and what are its main advantages and disadvantages?
34. What is the waterfall model, and what are its main advantages and disadvantages?
35. What is the spiral model, and when might it be most appropriate to use?
36. What is the iterative and incremental development model, and when might it be most appropriate to use?
37. What is requirement analysis in software engineering, and why is it important?
38. What is a software requirement specification (SRS), and what information Does it typically include?
39. How can requirements be validated and verified during the software Development process?
40. What is traceability in software requirements, and why is it important?

41. What are some challenges that can arise during requirement analysis and specification, and how can they be addressed?
42. What is requirement prioritization, and how can it be used to manage stakeholder expectations?
43. How can requirements be managed and tracked throughout the software development lifecycle?
44. What are software metrics and why are they important?
45. What are the different types of software metrics?
46. How do you choose the right software metrics to measure?
47. What is the difference between leading and lagging software metrics?
48. What is the difference between quantitative and qualitative software metrics?
49. How do you measure the effectiveness of a software metric?
50. What are some challenges in using software metrics effectively?
51. What are some common software metrics used in agile development?
52. How can you use software metrics to improve software quality?
53. How can software metrics be used to optimize software development processes?
54. What is software project planning and why is it important?
55. What are the steps involved in software project planning?
56. How do you estimate project timelines and costs during software project planning?
57. What is a project charter and why is it important in software project planning?
58. What is a project scope statement and why is it important in software project Planning?
59. How do you identify and manage project risks during software project planning?
60. What is a project schedule and how is it created during software project planning?
61. How do you measure and track project progress during software project planning?
62. What is project scheduling and why is it important?
63. What are the steps involved in project scheduling?
64. What is critical path analysis and how is it used in project scheduling?
65. How do you estimate task durations during project scheduling?
66. How do you manage resource availability during project scheduling?

67. What is a project baseline and why is it important in project scheduling?
68. How do you measure and track project progress during project scheduling?
69. What is a good software design? Why is it needed in the software development process?
70. Distinguish between preliminary and detail designs. What documents should be product on completion of preliminary and detail designs.
71. Explain how the characteristics of good design contribute to product quality.
72. Mention the various design guidelines.
73. Illustrate principles underlying a good design.
74. Explain the following:
75. Decomposition Information Hiding
76. Control Hierarchy
77. Structural Partitioning
78. Stepwise Refinement
79. Data Structure
80. Illustrate abstraction as a design concept.
81. Consider a program containing many modules. If a global variable x must be used share data between two modules A and B. how would you design the modules to minimize coupling
82. Briefly explain the software design process.
83. What do you understand by the term functional independence in the context of Software design?
84. What do you mean by the terms cohesion and coupling in the context of software design? How are these concepts useful in arriving at a good design of a system?
85. What are the objectives of testing?
86. Write short note on any three:
87. Concept of Project Management
88. System Testing
89. Fourth Generation Technique
90. Differentiate between the white box testing and Black Box testing.
91. What do you understand by software testing?
92. When should you stop the testing process?
93. Is it possible to achieve 100% testing coverage? How would you ensure it?
94. What are unit testing and integration testing?

95. Describe Black Box and White Box testing stating various strategies incorporated while performing each.
96. Can we do system testing at any stage?
97. What is a test plan and what does it include?
98. What is verification and validation in software testing?
99. What is a test case?
100. What are the phases involved in software testing life cycle?
101. Describe the benefits of automation testing.
102. What is the procedure to resolve issues during software testing?
103. What is Risk Analysis?
104. What is Risk? Explain the classification of risk management.
105. Explain the principles of Risk Management.
106. How Risk can be controlled during software development process
107. Define Software Reliability. State the difference between software and hardware reliability.
108. What are the factors that affect software reliability?
109. What is software metrics? Explain different types of software metrics.
110. What are the strategies used to achieve safety during risk in software development?
111. Explain the different types of maintenance that a software product might need.
112. Why is software maintenance required?
113. What is Software reverse engineering? Why is it required?
114. What is Software Reengineering? Why is it needed?
115. What are the challenges in software maintenance?