

PART – A

1. List and explain the key characteristics of software.
2. Describe the key elements of requirement specifications.
3. Provide examples of coding standards or conventions commonly used in a specific programming language, like Python or Java
4. How do you resolve conflicts in Git when merging or rebasing branches?
5. How do you select an appropriate design pattern for a given problem or scenario?
6. Discuss the importance of testing for expected errors in unit testing.
7. What is agile process? Mention its characteristics.
8. Explain pairwise testing and its advantages in reducing the number of test cases.
9. What are the prerequisites for implementing continuous integration in a software development project?
10. Provide an example of a use case where Ansible can be beneficial in a software project.
11. Explain Sashimi Model.
12. Explain COCOMO Estimation Model.
13. Explain Literate Programming.
14. Explain Conformance Quality. What are the techniques used for ensuring (3) conformance quality?
15. Explain Unit testing.
16. What is Anti-Patterns?
17. Explain Defect Life Cycle
18. What is Regression Testing?
19. Explain Version Control System? What is the purpose of using it?
20. Explain the benefits of Software Configuration Management?
21. What is the need for software engineering?
22. Write a short note on project estimation methods in software development.
23. What are the basic concepts of version control system?
24. How software quality is measured?
25. Define the concept of antipatterns.
26. What are assertions?
27. Define product backlog.
28. What is the relevance of software testing?
29. What is the purpose of software configuration"
30. What is test automation?
31. How do you measure software reliability?
32. What is COCOMO estimation model?
33. What are the four dimensions of software quality?
34. Explain the three states in which files reside in Git.
35. What are the characteristics of a design class?
36. Write about any three key classes in xUnit architecture.
37. List out the steps in agile methodology.
38. Explain decision table testing with suitable diagram.
39. What are the benefits of configuration management?
40. Distinguish between build automation and deployment automation

PART – B

Module I

1. Describe the typical phases in the software development life cycle.
2. Discuss the important concepts in software requirements engineering.
3. Module I Explain Prototype. Also its types, advantages and disadvantages
4. What are the characteristics of a Software?
5. With an example, explain requirement specification.
6. Explain predictive and adaptive waterfall models. How is it different from model in which development phases overlap each other?
7. Draw a waterfall model and explain the life cycle of a software system.
8. Prepare a basic software requirement specification for basic library system.

Module II

1. Compare and contrast literate programming and traditional code documentation approaches.
2. How can you set up Git on your local machine and configure it for your projects?
3. Explain how to clone a Git repository.
4. Explain the four dimensions of quality.
5. Explain the differences between "git fetch" and "git pull". How can conflicts be resolved in git?
6. Explain the core operations in Git Version Control System to manage a software project. Clone a repository using Git
7. Explain how to view the commit history in Git. Write the syntax of the commands used.
8. What is a Git repository? Explain the process of Cloning a Git repository

Module III

1. Describe the concepts of encapsulation, inheritance, and polymorphism in OOP.
2. Explain the concept of anti-patterns and why they should be avoided in software design.
3. Explain xUnit architecture and write example for testing using any frame work.
4. What in a Design pattern? Explain how to select a design pattern for your problem.
5. Write a short note on creational design pattern.
6. Illustrate the importance of writing tests with assertions.
7. What are single condition tests and expected error tests in Unit Test? Explain
8. Explain the concepts of Anti-pattern.

Module IV

1. Explain the concept of black-box testing and its various techniques.
2. With a neat diagram, explain scrum framework.
3. Explain Scrum framework.
4. Explain the principles of software delivery.
5. With a neat diagram, explain the scrum framework.
6. Explain the relevance of bug life cycle with a neat diagram.
7. What is blackbox testing? Explain.
8. Explain the meetings involved in scrum software development

Module V

1. Describe the process of managing build and deployment environments in a software development project.
2. Discuss the best practices for incorporating test automation into your software development process.

3. Explain the essential practices that should be enforced on Continuous Integration (CI) teams
4. Explain Black-Box testing and White-Box testing. Give any two techniques for each testing type.
5. Write a short note on the strategies involved in continuous integration.
6. What is CI/CD pipeline? Write the principles of software delivery.
7. With a neat diagram, explain deployment pipeline and the various stages of deployment pipeline.
8. Define continuous integration. Explain the essential practices required for continuous integration.