ADVANCED SOFTWARE ENGINEERING

Part A:3 marks each Part B: 6 marks each

MODULE 1

PART A

1. What is the need for software engineering?
2. Write a short note on project estimation methods in software development.
3. How do you measure software reliability?
4. What is COCOMO estimation model?
5. Explain Sashimi Model.
6. Explain COCOMO Estimation Model.
7. List out the main characteristics of a good software.
8. Write the advantages of prototyping development model.

PART B

1. Explain predictive and adaptive waterfall models. How is it different from model in which development phases overlap each other?
2. With an example, explain requirement specification.
3. Draw a waterfall model and explain the life cycle of a software system.
4. Prepare a basic software requirement specification for basic library system.
5. Explain Prototype. Also its types, advantages and disadvantages.
6. What are the characteristics of a Software?
7. Explain COCOMO. How can you estimate a software project using COCOMO II model?
8. Spiral model follows a risk-driven approach to help project teams decide on what development approach to take for various parts of the project. Justify the statement.

MODULE 2

PART A

1. What are the basic concepts of version control system?
2. How software quality is measured?
3. What are the four dimensions of software quality?
4. Explain the three states in which files reside in Git.
5. Explain Literate Programming.
6. Explain Conformance Quality. What are the techniques used for ensuring conformance quality?
7. Differentiate git pull and git fetch commands.
8. Write a note on literate programming.

PART B

1. Explain the core operations in Git Version Control System to manage a software project. Clone a repository using Git
2. Explain the differences between "git fetch" and "git pull". How can conflicts be resolved in git?
3. What is a Git repository? Explain the process of Cloning a Git repository.
4. Explain how to view the commit history in Git. Write the syntax of the commands used.
5. Explain how to clone a Git repository.
6. Explain the four dimensions of quality.
7. Illustrate the core operations in Git version control system to manage a software project in local system and remote server.
8. Explain the four dimensions of software quality.

MODULE 3

PART A

1. Define the concept of antipatterns.
2. What are assertions?
3. What are the characteristics of a design class?
4. Write about any three key classes in xUnit architecture.
5. Explain Unit Testing.
6. What are Anti-Patterns?
7. Explain the four essential elements of design pattern.
8. A design pattern may turn into an antipattern. Justify the statement.

PART B

1. Write a short note on creational design pattern.
2. Illustrate the importance of writing tests with assertions.
3. Explain the concepts of Anti-pattern.
4. What are single condition tests and expected error tests in Unit Test? Explain
5. What is a Design Pattern? Explain how to select a design pattern for your problem.
6. Explain xUnit architecture and write an example for testing using any framework.
7. Illustrate the architecture of xUnit framework with a neat diagram.
8. How design pattern is useful in software development? Explain in detail about different types of design patterns.

MODULE 4

PART A

1. Define product backlog.
2. What is the relevance of software testing?
3. List out the steps in agile methodology.
4. Explain decision table testing with suitable diagram.
5. Explain Defect Life Cycle.
6. What is Regression Testing?
7. Write any three software-testing principles.
8. Distinguish between black box testing and white box testing.

PART B

1. With a neat diagram, explain the scrum framework.
2. Explain the relevance of bug life cycle with a neat diagram.
3. Explain the meetings involved in Scrum software development methodology.
4. What is Black-box testing? Explain
5. Explain the Scrum framework.
6. Explain Black-Box testing and White-Box testing.
   Give any two techniques for each testing type.
7. Define agility. Explain various agile design principles.
8. Write short notes on equivalence class testing and control flow testing.

MODULE 5

PART A

1. What is the purpose of software configuration?
2. What is test automation?
3. What are the benefits of configuration management?
4. Distinguish between build automation and deployment automation.
5. Explain Version Control System. What is the purpose of using it?
6. Explain the benefits of Software Configuration Management.
7. Write down the principles of software delivery.
8. How dependencies are managed in software configuration management.

PART B

1. Write a short note on the strategies involved in continuous integration.
2. What is C/CD pipeline? Write the principles of software delivery.
3. Define Continuous Integration. Explain the essential practices required for CI.
4. With a neat diagram, explain the Deployment Pipeline and the various stages of the deployment pipeline.
5. Explain the principles of software delivery.
6. Explain the essential practices that should be enforced on Continuous Integration (CI) teams.
7. Explain the essential practices of continuous integration.
8. With a neat diagram, explain the architecture of deployment pipeline.