Questions:

1. In a Scrum-based software development project, the Product Owner has defined the following user stories for an e-commerce application:

- As a user, I want to log in securely so that I can access my account.
- As a user, I want to search for products by category to find items easily.

i. Create a product backlog for these user stories by breaking them into tasks.

ii. Describe how the development team can prioritize these user stories during a Sprint Planning meeting, considering value to the customer and technical feasibility.

iii. Illustrate how these tasks will be tracked using a Scrum board. Use proper terms like "To Do," "In Progress," and "Done." **(6)**

2. A software development team is about to start a project for a new innovative product. The project has several high-risk components due to its novelty, and there's uncertainty regarding the client's future needs. The client is open to iterative changes, but the team must ensure that the software evolves in a manageable, cost-effective way.

I. Considering the high risks and the evolving nature of the client's needs, discuss how the Spiral, Agile, and Extreme methodologies address risk management and adaptability. Which methodology would be the most suitable for a project with significant risk and evolving requirements, and why? **(6)**

3. A company is working on two different projects. Project A has well-defined requirements and a strict deadline, while Project B has evolving requirements with an uncertain timeline and continuous customer feedback. Both projects involve high stakes, and the team must decide which development methodology to use.

I. Compare and contrast the Waterfall, Agile, Extreme, and Spiral development models. Based on the characteristics of both projects (Project A and Project B), which methodology would best suit each? Support your answer with a detailed analysis of how each methodology would address the specific needs of the projects, considering factors such as predictability, customer collaboration, and risk management. **(6)**

4. Explain the principles of software engineering ethics, highlighting the issues related to professional responsibility. Discuss how the ACM/IEEE Code of Ethics guides ethical decision-making in software engineering practices. **(4)**

5. Given the story of the Airport Reservation System, identify at least five **functional** and five **non-functional** requirements for the system. In your answer, explain how each requirement contributes to the overall performance, usability, and security of

the system. Consider factors such as performance, user experience, and system maintenance in your discussion. **(4)**

6. Illustrate and explain the V-model of testing phases in a plan-driven software process, detailing the relationships between development activities and corresponding testing activities. **(4)**
7. Explain the process of prototype development in software engineering. Discuss the key stages involved in creating a prototype and how it helps in refining software requirements. Analyze the benefits of using the prototyping model, particularly in terms of user feedback, risk reduction, and iterative development. **(4)**
8. Explain the process improvement cycle in software engineering and describe its key stages. Name and explain some commonly used process metrics, highlighting how they help in monitoring and improving software processes. **(4)**
9. Explain the Software Engineering Institute Capability Maturity Model (SEI CMM) and its five levels of capability and maturity. Analyze how each level contributes to improving the software development process and organizational performance. **(4)**
10. Describe the core principles of agile software development methods. Analyze how these principles are applied in different software development environments, and assess the benefits and challenges of using agile methods in various project types and organizational settings. **(4)**
11. Draw the release cycle of (Extreme Programming (XP) and explain the influential programming practices. **(4)**
12. A local library wants to create a digital system to manage its operations. The system will track books, members, and borrowing activities. Each book has attributes like title, author, ISBN, and genre. Members have attributes such as name, membership ID, and contact details. When a member borrows a book, the system records the borrowing date, return due date, and return status. The library also wants to maintain a catalog of overdue books and their respective fines.

I. Using the scenario of a digital library management system, design an Entity-Relationship Diagram (ERD) to represent the entities (e.g., books, members, borrowing activities) and their relationships. Clearly explain the attributes of each entity and how they are interconnected. **(6)**

13. What is called Testing? Differentiate between Validation and Verification. **(4)**
14. Design a layered architecture model for an online judge system, identifying the key layers (e.g., presentation, application, business logic, and data). Explain the responsibilities of each layer and analyze how this architecture ensures scalability, maintainability, and efficient performance. **(4)**
15. Draw a DFD (Level-0 and Level-1) and UML Use Case Diagram for a Hospital Management System. A hospital management system is a large system that includes

several subsystems or modules that provide various functions. Your UML use case diagram example should show actors and use cases for a hospital's reception.

I. **Purpose**: Describe major services (functionality) provided by a hospital's reception.

II. Consider the **Scenario** below:

III. The Hospital Reception subsystem or module supports some of the many job duties of a hospital receptionist. The receptionist schedules patient's appointments and admission to the hospital and collects information from the patients upon patient's arrival and/or by phone. The patient who will stay in the hospital ("inpatient") should have a bed allotted in a ward. Receptionists might also receive patients' payments, record them in a database and, provide receipts, file insurance claims and medical reports. **(5)**

16. Consider the following UML Sequence diagram and design the UML diagram for the same scenario.

I.

II. Now Draw a UML Class diagram for the scenario depicted by the Sequence Diagram. **(5)**

17. We know that Quality Assurance (QA) is not just Quality Control (QC). For example, QA is process-oriented, and QC is product-oriented. Now, suppose you are said to get a straight explanatory description of it along with the Differences and Impediments between and to QA and QC. **(5)**

18. Do you think the goal of QA is just to find the bugs as early as possible? The goal of the QA is to find the bugs as early as possible and make sure they get fixed. Quality Assurance was introduced after World War II when the weapons were tested before they came into action. Explain the role of Quality Assurance (QA) at each phase of the Software Development Life Cycle (SDLC). **(5)**

19. Explain the Rapid Application Development (RAD) model in software engineering. Discuss its key phases, principles, and advantages. Analyze how the RAD model supports faster delivery of software solutions while maintaining quality and user satisfaction. **(5)**

20. White box testing consists of code coverage and a data coverage method. Consider the following decision (e.g. if, switch, while etc.) and make one test for each side of each decision using a table with column caption Decision, x input and y input. Then implement JUnit test class that tests each decision described in the table using Java (follow the sample in Python) **(6)**

I.
```
int x, y;
x = c.readInt();
y = c.readInt();
```

```
if (y == 0)
    c.println("y is zero");
else if (x == 0)
    c.println("x is zero");
else {
    for (int i = 1; i <= x; i++) {
        if (i % y == 0)
            c.println(i);
    }
}

// JUnit test class that tests each decision described in the table
using Python as follows, do the same in Javaimport unittest

class DecisionTest(unittest.TestCase):

    def setUp(self):
        # Mocking the behavior of `c.println` using a list to
collect output
        self.output = []

    def println(self, message):
        # Simulates the `c.println` function
        self.output.append(str(message))

    def process(self, x, y):
        # The translated logic of the Java code
        if y == 0:
            self.println("y is zero")
        elif x == 0:
            self.println("x is zero")
        else:
            for i in range(1, x + 1):
                if i % y == 0:
                    self.println(i)

    def test_y_is_zero(self):
        self.output.clear()
        self.process(5, 0)
        self.assertEqual(self.output, ["y is zero"])

    def test_x_is_zero(self):
        self.output.clear()
        self.process(0, 3)
        self.assertEqual(self.output, ["x is zero"])

    def test_loop_does_not_run(self):
        self.output.clear()
        self.process(0, 2)
        self.assertEqual(self.output, [])

    def test_numbers_divisible_by_y(self):
        self.output.clear()
        self.process(4, 2)
        self.assertEqual(self.output, ["2", "4"])

    def test_numbers_not_divisible_by_y(self):
        self.output.clear()
        self.process(4, 3)
        self.assertEqual(self.output, ["3"])

    def test_edge_case_y_negative(self):
        self.output.clear()
        self.process(5, -2)
        self.assertEqual(self.output, ["2", "4"])

    def test_edge_case_x_negative(self):
        self.output.clear()
        self.process(-3, 2)
        self.assertEqual(self.output, [])

if __name__ == "__main__":
    unittest.main()
```

21. Black Box Unit testing is earlier and more precise than Black Box System testing - it can find errors very early, even before the entire first version is finished. Now, Consider the production codes that need function testing. Suppose you have JUnit 4 API in your IDE and you are said to develop test codes for these production codes

showing the application of Exception, Setup Function and Timeout Rule. How do you solve it? **(5)**