ICS 2312: Software Systems Development - Exam Answers

Question 1

Question 1

a) i. Definitions:

- Object: A self-contained entity combining data and methods. An instance of a class.

- Use Case: A user interaction scenario describing a system's functional requirement.

ii. Three non-functional requirements:

- Performance: e.g., the system should respond in <2 seconds.

- Scalability: e.g., system should support 10,000 users.

- Security: e.g., enforce login with password encryption.

b) i. Use case modeling is used to gather and describe user requirements.

- Stage: Requirements Analysis.

- Output: Use Case Diagram, Use Case Descriptions.

ii. Abstract vs Concrete Classes:

- Abstract: Cannot be instantiated; used to define templates. Example: `abstract class Animal`.

- Concrete: Fully implemented classes. Example: `class Dog extends Animal`.

c) Responsibilities of each step:

- Specify: Define what system will do.

- Prototype: Mockups or simulations to verify requirements.

- Design: Define architecture, components, and interfaces.

- Deploy: Install system and monitor in production.

d) Vending Machine State Diagram:

- States: Idle -> Ready -> Dispensing

- Transitions: Insert Coin -> Push OK -> Deliver Item

Question 2

Question 2

a) Use Case Diagram for Retailer System:

Actors: Retailer, Supplier, Customer.

Use Cases: Check stock, Order goods, Arrange goods, Sell goods, Prepare bill, Make payment.

Relationships: Retailer includes Check Stock; Supplier extends Order Goods; Customer includes Make Payment.

b) Rational Unified Process (RUP) Phases:

1. Inception: Identify business case and scope.

2. Elaboration: Refine requirements and define architecture.

3. Construction: Develop and implement features.

4. Transition: Deploy and train users.

Graph representation: Overlapping phases on a timeline, showing effort distribution.

ICS 2312: Software Systems Development - Exam Answers

Question 4

Question 4

a) Sequence vs Collaboration Diagrams:

- Sequence Diagram: Shows message flow over time. Use when order matters.

- Collaboration Diagram: Emphasizes object structure and relationships. Use when focusing on who interacts with whom.

b) Cohesion:

- Cohesion: Degree to which tasks within a module relate.

- Types:

  - Functional Cohesion: Single well-defined task.

  - Sequential Cohesion: One task's output feeds into the next.

c) Activity Diagram for Order Processing:

- Start -> Receive Order -> Submit Credit Info -> [Decision: Valid?] -> Send Messages to Finance & Stores -> Check Plan -> Apply Bonus Points -> End

d) Real-Time Systems:

i) Classifications:

- Hard Real-Time: e.g., Pacemakers (deadline must not be missed).

- Soft Real-Time: e.g., Online games (late is undesirable but tolerable).

ii) Three Necessary Conditions:

- Determinism: Predictable behavior.

- Timeliness: Tasks must finish on time.

- Fault Tolerance: Should handle failures gracefully.