**True/False Questions**

1. **False**: Boundary objects can usually communicate with both controllers and actors, but they can also interact with entity objects in certain scenarios.
2. **False**: A use case typically has clear boundaries that define its scope and the interaction between the system and actors.
3. **True**: Controllers are designed to manage interactions between boundary objects, entity objects, other controllers, and actors.
4. **False**: Entity objects primarily communicate with other entity objects or controllers, but they do not usually communicate directly with boundary objects.

**Short Answer Questions**

1. **What is a requirement?**
   * A requirement is a specification of what a system should do or a constraint on the system's operation. It defines the functionalities, performance, and attributes that a system must have to meet the needs of its users or stakeholders.
2. **Why is writing use cases an iterative process?**
   * Writing use cases is iterative because as the understanding of the system evolves, new use cases may be discovered, or existing use cases may need to be refined. Iteration allows for continuous improvement and adaptation to ensure that all user interactions are accurately captured.
3. **Explain actor, goal, and scenario:**
   * **Actor**: An actor represents any entity (human or system) that interacts with the system. Actors can be users, external systems, or devices.
   * **Goal**: A goal is the objective or outcome that the actor wants to achieve through their interaction with the system.
   * **Scenario**: A scenario is a specific sequence of actions and interactions between the actors and the system that leads to the achievement of the actor's goal.
4. **What are the classifications of objects discovered?**
   * Objects discovered during analysis and design can be classified into:
     + **Boundary Objects**: Objects that interface with actors, such as user interfaces, APIs, or input/output devices.
     + **Entity Objects**: Objects that represent business entities or data within the system, typically corresponding to tables in a database.
     + **Control Objects**: Objects that manage the flow of control between boundary and entity objects, handling the business logic and process control.