1. (12 points) Explain the similarities and differences between **two specific software process methodologies**. Identify the pros and cons of each approach, and specify what types of projects should apply each of these methodologies.

2. (15 points) Complete a given unfinished UML Use Case diagram by drawing and labeling the associations between the actors and the use cases in the diagram to represent the interactions that occur in a particular scenario.

3. (10 points) Use a particular software system scenario to answer certain specific questions regarding its functional and non-functional requirements.

4. (16 points) Given a specific unfinished UML class diagram representing a particular system. Using the appropriate symbols, transform all of the associations in this diagram that should be generalizations, aggregations, or compositions.

5. (10 points) Use a pair of UML class diagrams to explain a specific software design principle (e.g., loose coupling, high cohesion, information hiding, separation of concerns, design for change).

6. (15 points) Consider two specific procedures for accomplishing a particular task. Use two provided sets of lifelines to complete sequence diagrams to model each of these approaches.

7. (12 points) Apply a specific one of the GRASP design patterns (i.e., Controller, Expert, and Creator) to explain how a certain aspect of a particular software system should be handled.

8. (10 points) Explain the fundamental differences between **three different types of software testing**, emphasizing the distinctions in **who** does the testing, **when** they do it, and **why** they do it.