

Student Name \_\_\_\_\_ Student Number \_\_\_\_\_

Concordia University  
Department of Computer Science and Software  
Engineering

COMP 6471 Software Design Methodologies  
Quiz 1

**Course:** COMP 6471/4 Section NN

**Date:** Monday, 13 February 2006

**Time:** 18:00 – 19:30

**Instructor:** Greg Butler

**Coordinator:** Greg Butler

**Special Instructions and Information**

- This is a closed-book examination. You may not use any materials other than normal writing implements.
- Pocket calculators are not allowed.
- Answer all four questions. The marks for each question are indicated. The total marks are 60.
- Answer questions in the space provided on the examination paper.

**Question 1.** [20 marks]

For each of the following statements, indicate whether the statement is true or false by circling T or F respectively.

- (1) An object is an instance of a class. **T**
- (2) A class is characterized by its state, its behaviour, and its identity. **F**
- (3) An abstract class cannot be directly instantiated. **T**
- (4) There can be no instances of an abstract class. **F**
- (5) A use case is a class. **T**
- (6) An event is the specification of a significant occurrence that has a location in time and space. **T**
- (7) An interface is a collection of operations. **T**
- (8) A class has a name, attributes, and operations. **T**
- (9) Aggregation is a special kind of composition. **F**
- (10) Associations are inherently bidirectional, but can be made directional by placing arrowheads at one end of the line that represents the association. **T**
- (11) The timeline in a sequence diagram is measured in units of one millisecond. **F**
- (12) The messages in a collaboration diagram must have sequence numbers. **T**
- (13) A use case may be described by one or more scenarios. **T**
- (14) A scenario may be described by any one of the interaction diagrams. **T**
- (15) It is possible to have a generalization relationship between two actors. **T**
- (16) Flow of information in UML can be described using either messages or object flows. **T**
- (17) A statechart diagram describes a scenario. **F**
- (18) Association roles qualify an association and indicate how to access the collection of associated objects. **F**
- (19) An activity diagram is specific to a single object, and describes an operation for that object in terms of activities. **F**
- (20) An interface can be realized by the operations of a class. **T**

**Question 2.** [10 marks]

For each of the following statements, indicate whether the statement is true or false by circling T or F respectively.

1. A good pattern is a named and well-known problem/solution pair that can be applied in new contexts, with advice on how to apply it in novel situations and discussions of trade-offs, implementations, variations, and so forth. **T**
2. RDD leads to viewing an OO design as a community of collaborating, responsible objects. **T**
3. The basic types of responsibility are doing, knowing, owning, and creating. **F**
4. The UML defines a responsibility as a contract or obligation of a classifier. **T**
5. The Factory design pattern is one application of the Creator principle. **T**
6. The main role of the High Cohesion GRASP principle is to support the Low Coupling principle. **F**
7. The problem that the Low Coupling GRASP principle addresses is to reduce the impact of change. **T**
8. All solutions proposed by the Controller GRASP principle are variations of a facade. **F**
9. The facade design pattern is an application of the Low Coupling GRASP principle. **T**
10. The motivation for the Information Expert GRASP principle is not low coupling, rather the distribution of responsibility for information. **F**

**Question 3.** [10 marks]

Complete each sentence below by supplying the missing word(s) in the space provided.

1. A critical ability in OO development is to skilfully assign **responsibilities** to software objects.
2. During object-oriented **analysis** there is an emphasis on finding the objects in the problem domain.
3. **Inception** in one sentence: Envision the product scope, vision, and business case.
4. A **scenario** is a specific sequence of actions and interactions between actors and the system.
5. **Use cases** are text stories, widely used to discover and record requirements.
6. A **use case** is a collection of related success and failure scenarios that describe an actor using a system to support a goal.
7. Three kinds of actors are **primary, supporting, off-stage**.
8. Actors are **roles** played not only by people, but by organizations, software, and machines.
9. A **domain model** is a visual representation of conceptual classes or real-situation objects in a domain.
10. The **extension** of a conceptual class is the set of examples to which the class applies.

Write sentences.

0 points if off-topic, no answer, or more wrong things than right things

1 point for something correct

2 points for substantial answer

3 points for good answer

4 points for proper terminology plus awareness of issues

5 points for complete insightful answer

**Question 4.** [20 marks]

- (a) **Why** does the Unified Modeling Language (UML) and object-oriented design (OOD) use models?

recording, communicating, understanding, analysis, reasoning

abstraction, emphasize important info, omit unimportant info

view, multiple views, multiple kinds of views, match perspective of stakeholder, match info important for issue

visual

blueprint

- (b) **Explain** three kinds of classes that might occur in a domain model.

See Table 9.1 Conceptual Class category List

- (c) What is RDD? What does it have to do with collaboration?

**Responsibility-Driven Design**

RDD is design by assigning responsibilities amongst objects/classes/...

guided by GRASP principles

an object fulfils responsibility by collaborating with other objects that have information or responsibility that is required

- (d) What is a **design** pattern? **Explain** the Command pattern and its **importance** to **your** project.

design pattern is problem/solution pair in context for a design problem ...

Command pattern separates request for action from invocation of action ...

Importance ...