

# CS314 2012 Test 1 Study Guide

My Tests typically have the following format:

**Part 1:** Multiple-choice questions (about 10 questions)

**Part 2:** Short answer questions (e.g., “Why should instance variables be private in a class?”)

**Part 3:** Modeling Questions: In this section I may give a partial UML models and may require answers to one or more of the following: (1) questions based on the information presented in the models (e.g., If I give you an example of a class model for an Order Processing System I may ask a question such as “Must an order have at least one order item?” which requires you to look at the multiplicities on the association between the Order and OrderItem classes) and (2) questions that require you to extend the models to reflect additional information.

**Study my course slides up to and including Week 5 lectures. Also read Chapters 1, 2, and 5.** Note that the textbook has examples of design class models with associations with no names and no multiplicities – *for my course and tests all associations must have meaningful names (I’ll make sure that meaningful names can be easily found in the models I give on the test!) and must have multiplicities. Please follow all the design modeling styles and guidelines I give in my slides.*

Students should be able to answer the following questions or understand the following concepts before taking Test 1:

## ***Intro to SE***

1. Understand the differences between programming-in-the-large and programming-in-the-small.
2. Understand the differences between modeling-in-the-small and modeling-in-the-large.
3. Understand the role of modeling in engineering software.

## ***OO Programming Review***

1. What is the relationship between a class and an abstract data type?
2. What is polymorphism? What is dynamic binding?
3. Understand the difference between overriding and overloading.
4. Distinguish between good and bad uses of inheritance.
5. Know the difference between inheritance and “implements” relationships in Java.
6. Understand Liskov’s Substitutability Principle (LSP) and its relationship to inheritance structures.
7. What is the factory method and how can it be used to improve program design?

## ***Design Class Diagrams***

1. What is a design class diagram?
2. How is a design class represented in UML?

3. Why are references to other objects not shown as attributes in class models?
4. When do you model a concept as an attribute rather than as a class?
5. Understand when to model a property as an attribute or as an association.
6. What is a role?
7. What is an association class and how is it represented in UML?
8. How can a class diagram with association classes be converted to an equivalent class diagram without association classes?
9. What are the navigability symbols and what do they represent?
10. What does it mean when an association in a class diagram does not have a navigability symbol?
11. How is inheritance represented in a design class diagram?
12. What is multiple inheritance?
13. Understand when to use associations instead of inheritance.
14. Given a design class diagram you should be able answer questions about the information it captures.
15. Given a program you should be able to draw a class diagram of its structure.
16. Given a textual description of a problem you should be able to draw a class diagram.
17. What is an object diagram?
18. Understand how a class diagram can be used to determine of an object model is valid or not.

### ***Basic Design Principles***

1. Distinguish between information hiding and encapsulation.
2. What is a (class) responsibility and how is it manifested in an OO design?
3. What are some of the benefits of properly distributing responsibilities across classes?
4. What are some of the bad things that can happen if responsibilities are not properly distributed in a design?
5. Identify and explain the importance of the following responsibility assignment principles: Cohesion, Expert, Creator, Coupling, Controller