

CS4233-A04 Midterm Exam Study Guide

If you have read the assigned readings, done the homework assignments, attended the lectures and studied the class notes, you should be adequately prepared for the exam. These notes identify some of the specific areas that you might encounter on the exam.

General O-O knowledge

- You should be able to define *encapsulation, inheritance, polymorphism, class, object*.
- How does Java support encapsulation?
- How does Java support multiple inheritance? How is it different than C++?
- When talking about objects, what do the terms *state, behavior, and identity* mean? How are they represented in Java?
- What is cohesion? What is coupling? Which is more desirable for each, high or low? (Low coupling, high cohesion) Why?
- What are the different ways of cloning an object? How do you do it? What are the consequences of the different types of cloning, or not using cloning?

UML

- You should be able to read and write the following type of diagrams: class, sequence, collaboration, and state.
- You should be able to describe associations, multiplicity, association names, dependency.
- What is aggregation and composition? What relationships do they model?
- In UML diagrams, how can you tell whether a rectangle with a name in it represents an object or class? (Object names are underlined)
- Given a use case you should be able to construct one or more sequence diagrams.
- You should be able to convert a sequence diagram to a collaboration diagram and vice-versa.
- How do you show inheritance? Interface realization?
- Given a description of a class, you should be able to construct a state diagram. Given a state diagram, you should be able to answer questions about the behavior of the class.

Analysis & Design (general principles)

- What is layering? Give an example?
- Why is it considered poor design to have objects in a layered architecture communicate with objects that are more than one layer away? What are the possible effects of this? (Higher coupling of layers. Harder to replace a single layer.)
- What is the difference between analysis and design?
- What goes into an analysis model?
- What are the three type of analysis classes (boundary, control, entity) and what is the purpose of each?
- What is the purpose of packages (in the general, not just Java sense)?
- Given a problem description or vision, you should be able to perform a textual analysis to identify analysis classes.
- What is the primary purpose of class design? (Assign responsibilities)
- You should be able to, given an analysis of a system, assign responsibilities to design classes.
- What is architecture?

Class Design

- What are static relationships? Give examples. Given a description, describe the static relationships between objects.
- What is the difference between inheritance and interface realization?
- What is delegation? When is it preferred over inheritance? What are the advantages and disadvantages?
- What is the state of an object? How do you describe state changes?
- What are accessors and mutators (getters and setters)? When should you use them?
- What makes a good accessor and mutator?
- When would you use an immutable object? Give an example? (One example is the remote mower status, although I'd explicitly tell you that you can't use that as the example.)

CS4233-A04 Midterm Exam Study Guide

- What is LCOM? How is it computed (in a general sense)?
- Given a description of a class implementation, determine if it is cohesive and, if not, fix it (by creating multiple, cohesive classes)?
- Given a class or interface, evaluate it for completeness and consistency. Identify methods that are essential and those that are convenience.

Testing

- You should be able to write a good JUnit test.
- What is Test-First Programming (or Test-Driven Development)? What are the benefits that you are supposed to realize from using it? (Evolving design, 100% code coverage, regression suite built as you go).

Patterns

- What is the “rule of 3”?
- What is a pattern? How is it different than a template?
- Where did patterns come from?
- How do you describe a pattern?
- What is delegation? Show delegation using a UML diagram? Give an example of delegation?
- What is the Iterator pattern? Show the Iterator pattern using a UML diagram? Give an example? What are the consequences of applying the Iterator?
- What is the Observer pattern? How is it used? Show the Observer using a UML diagram? Be able to implement the pattern in Java.

Threads

- What is a thread? How is a thread different from a process?
- What does it mean to say that something is *thread safe*?
- How do you pause threads and start them again?
- Given a situation, design an appropriate thread strategy for implementing the situation.