Midterm Exam: Review Topics

Back

Object-Oriented Programming

- What are coupling and cohesion?
- How are state and behavior related to objects?
- What are Abstraction? Encapsulation? Inheritance? Polymorphism?

Declarations, References, and Instances

- Compare the use of classes and interfaces as types in Java.
- What is meant by static type? dynamic type?
- What is type compatibility?
- What are scope, visibility, and lifetime?
- What are the visibility modifiers / access levels?
- What are other modifiers? When can they be used? (includes **static**, **final**, **abstract**)
- What is the effect of the **import** declaration?

Inheritance

- Describe the Is-A relationship.
- Describe the Has-A relationship.
- What is method overloading?
- What is method overriding?
- Can one overload constructors?
- Can one override constructors?
- What is dynamic dispatch?
- Describe the two uses for **super**.
- Describe the difference between abstract and concrete classes.
- Know how to write Java classes, both abstract and concrete (non-abstract).
- In overriding, how is the correct method chosen?
- In overloading, how is the correct method chosen?
- What is a Java interface?
- Know how to write a Java interface.
- Compare and contrast the use of abstract classes and interfaces. How are they similar? How are they different?
- Know the syntax for inheriting methods from classes, from interfaces.
- Can one write a class without any constructors?
- Can one declare a constructor **private**? How could such a constructor be used?

Unit Testing

- What is the standard package for unit testing in Java?
- What is a test case?
- How are test cases created? (annotated)
- Should test cases be large or small?
- Should testing be comprehensive?

- What kinds of test should be included? (three primary types)
- What do @Test, @Before, and @After do?
- How are they used?

Packages

- What is a Java package?
- How is a Java package related to the file system?
- How is a Java package related to the class path?
- What is a namespace?
- Do all Java classes exist in named packages?
- If no, where are the classes which are not in named packages?
- How is a package specified in Java source code?
- What is the impact / effect of the **import** declaration?

User Interface

- Describe the relationship between components and containers.
- Know the basic usage of JFrame, JButton, JPanel, JLabel. (You should know the methods that you used in the assignments and labs.)
- Describe the actions / use of a layout manager.
- Compare and contrast three basic layout managers (FlowLayout, GridLayout, BorderLayout).
- Be able to create simple user interface using JFrame, JButton, JPanel, JLabel, and the layout managers.
- Describe the use of paint / paintComponent. How are they similar? different?
- Describe the relationship of paint / paintComponent and the underlying windowing system.
- What is a Graphics object? How is it used?

Event-Driven Programming

- Contrast event-driven programming with algorithm-driven programming.
- Describe the primary features of Java v1.1 event handling.
    - Listeners
    - Adapters
    - Event objects
    - Event sources
- Be able to list some simple events and objects which may generate those events.
- Describe how listeners are associated with events. (cf: Observable - Observer pattern)
- Describe the use of inner classes in event handling. (not included on the exam: anonymous inner classes and accessing inner classes externally.)

Programming by Contract

- What is meant by "Programming by Contract"?
- What is a precondition? Postcondition? Invariant?
- Whose responsibility is it that postconditions be met?
- Whose responsibility is it that preconditions be met?
- When preconditions are not met, what should occur?
- What is an invariant?
- What is an assert?
- What is an exception?

Exceptions

- Define the term "exception".
- What does "the exception propogates" mean?
- When does an exception propogate?
- What does "report the exception" mean?
- Know the syntax to raise an exception.
- Know the syntax to handle an exception.
- Know the syntax to report an exception.
- Know the distinction between Java exceptions, errors, and run-time exceptions.
- Syntactically, what kind of "critter" is a Java exception? A Java error?
- What is the difference between checked and unchecked exceptions?
- Can two exceptions occur concurrently?

Collections

- What is a collection?
- What is an iterator?
- The standard Java2 libraries support three different kinds of collection-like objects. What are they? Which are true collections? Which is not?
- In which package would one find collections?
- How are these collections represented in the library?
- What are some standard implementations of these kinds of collections?
- Why are there multiple implementations?
- What is a ListIterator? How is it related to Iterator?
- What is a stack? A queue?

Back