CSCI 135 – Fundamentals of Computer Science I
Exam II Study Outline

I. Graphics and Audio
   A. File Input
      1. Scanner class
      2. try/catch
      3. Exceptions
   B. StdDraw
      1. Drawing simple shapes
      2. Drawing images
      3. Manipulating the output window (canvas)
      4. Animation
      5. Keyboard input
   C. StdAudio
      1. Playing sounds
      2. Manipulating sound files

II. Problem Decomposition
   A. Understand the Problem – Specification
      1. Looking at input and output first
   B. Work out the Logic – Design
   C. Convert it to Code - Programming

III. Methods
   A. Static Methods
      1. Parameters
         a. Pass by Value
      2. Return type
      3. Signature
         a. Overloaded Methods
      4. Variable Scope
   B. Flow of Control
   C. Calling a Method

IV. Programming Style
   A. Meaningful Names
   B. Comments
   C. Indentation
   D. Named Constants
   E. Whitespace
   F. Curly Braces

V. Testing and Debugging
   A. Preventing Bugs
      1. Write pseudocode (English-like) first
      2. Comment the tricky parts
3. Good coding style
   a. Variable names
   b. Break into manageable steps
   c. Indentation
   d. Watch loop bounds
   e. Listen to Eclipse feedback

4. Incremental development

B. Finding Bugs
   1. Add debug print statements
   2. Talk through the logic

C. Testing
   1. Software Quality
   2. Levels of Testing
      a. Unit
      b. Integration
      c. System (alpha) Test
      d. Acceptance (beta) Test

3. Black Box vs. White Box Testing

4. Equivalence Classes of Test Data
   a. Valid Inputs
   b. Invalid inputs
   c. Errors, exceptions and events
   d. Boundary conditions

VI. Exceptions
   A. Defending against bad input
   B. Handling unexpected events

VII. Classes
   A. Creating your own data types
      1. Classes
      2. Objects
      3. Instance variables (state)
      4. Instance methods (behavior)
      5. Constructors
      6. Arrays of objects

   B. this

   C. main methods for testing

VIII. Inheritance
   A. Advantages
      1. Allows code sharing – avoid repeated code!
      2. Store similar objects in same containers

   B. Subclasses and Superclasses
      1. super keyword
      2. Method Overriding
      3. Which method executes?

   C. Access modifiers