

# CSCI 135 – Fundamentals of Computer Science I

## Final Exam Study Outline

- I. Understanding of Everything from Before... especially...
  - A. Basic Programming Constructs
  - B. Problem Decomposition
    - 1. Understand the Problem – Specification
      - a. Looking at input and output first
    - 2. Work out the Logic – Design
    - 3. Convert it to Code - Programming
  - C. Object-Oriented Problem Decomposition
    - 1. Identifying Classes
    - 2. Identifying Information the Class Needs
      - a. Attributes, or Instance Variables
    - 3. Identifying Class Behaviors
      - a. Methods
- II. 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. self keyword
    - 3. Method Overriding
    - 4. Which method executes?
- III. Designing Data Types
  - A. Data Encapsulation Model
    - 1. Classes
    - 2. Client(s)
    - 3. API (Application Programming Interface)
  - B. Data Encapsulation
    - 1. Getters (Accessors)
    - 2. Setters (Mutators)
  - C. Immutability
  - D. Checking for Equality
    - 1. Primitive Data Types
    - 2. Reference Data Types
  - E. Important Methods:
    - 1. Constructors
    - 2. Getters and Setters (Accessors and Mutators)
    - 3. Equality Checking: equals()
    - 4. Printable Representation: toString()
    - 5. All other behaviors
- IV. Exceptions
  - A. Defending against bad input

- B. Handling unexpected events
- 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 Idle/compiler 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
    - 5. Regression Testing