

CSCI 135 – Introduction to Computer Science I

Exam II Study Outline

I. Java Basics

- II. Data Types
- III. Error Types and Debugging
- IV. Conditionals
- V. Loops
- VI. Programming Style
- VII. Command Line Input
- VIII. Standard Input
- IX. Arrays
- X. Graphics and Audio

II. Static Methods

- A. Using Static Methods
 - 1. Static methods we've already used
- B. Defining our own Static Methods
- C. Concepts
 - 1. Return Data Type
 - 2. Parameters (and data types)
 - 3. Modularity
 - 4. Calling a Method
 - 5. Flow of Control
 - 6. Variable Scope
- D. Terminology
 - 1. Access Modifier
 - 2. Return Type
 - 3. Parameters/Arguments
 - a. Pass by Value
 - b. Pass by Reference
 - 4. Method Name
 - 5. Return Statement
 - 6. Method Signature
 - 7. Overloading

III. Libraries and Clients

- A. Terminology
 - 1. Library
 - 2. Client
 - 3. API
 - 4. Implementation
- B. Unit Testing
- C. Modular Programming
- D. Dependency Graph

IV. Recursion

A. Mathematical Induction

1. Base Case
2. Induction Step

B. Recursion vs. Iteration

C. Things to Avoid

1. Missing Base Case
2. No Convergence
3. Stack Overflow

V. Classes and Objects

A. Classes as Data Type

B. Classes (Templates)

C. Objects (Instances of a Class)

1. Instance Variable – things they know
2. Instance Methods – things they do
3. Access Modifiers for both
4. Instantiating objects

D. Constructors

E. Arrays of Objects

F. Null Value

VI. Objects, Primitives and References

A. Primitive Type vs. Object Reference Type

B. Orphaned Objects

C. Garbage Collection

D. Aliasing

E. this

VII. Designing Data Types

A. Data Encapsulation

1. Hiding internal representation
2. Separating implementation from design specification
3. Client / API / Implementation
4. Access modifiers
 - a. public
 - b. private
 - c. (none) - default
 - d. protected
 - e. final

5. Immutability

B. Equality

C. Procedural Programming (verbs) vs. Object Oriented Programming (nouns)

D. Instance methods your class may need:

1. Constructor(s)
2. equals()
3. toString
4. Getters (Accessor Methods)
5. Setters (Mutator Methods)