

# SE1011 Detailed Course Objectives

Dr. Walter Schilling

Fall, 2011

You may bring one 8.5 x 11 sheet of paper with notes to the final exam.

## 1. Week 1

### (a) Writing Computer Software

- i. Describe the steps involved in creating and running a Java program
- ii. Describe the contents of source (.java) and class (.class) files
- iii. Explain what happens (at a high level) when a Java program is compiled
- iv. Explain what happens (at a high level) when a Java program is run
- v. Describe the difference between compilation and execution errors
- vi. Explain why a Java Virtual Machine (JVM) is required in order to run a Java program
- vii. Describe how bytecode makes Java programs portable
- viii. List the basic steps involved in software development

### (b) Algorithms and Design

- i. Define the term algorithm
- ii. Explain the motivation for doing design before coding
- iii. Make use of variables and operations to perform calculations
- iv. Construct and interpret flowcharts representing sequential, conditional, and looping structures
- v. Construct and interpret pseudocode representing sequential, conditional, and looping structures
- vi. Use flowcharts and pseudocode to describe algorithmic solutions to simple problems

## 2. Week 2

### (a) Primitive datatypes, Variables, Identifiers

- i. List the primitive types supported in Java: int, long, float, double, and char
- ii. Select the most appropriate primitive type to store a given piece of data
- iii. Use the assignment and compound assignment statements
- iv. Describe what happens in memory when a primitive variable is declared
- v. Describe what happens in memory when an object identifier (reference) is declared
- vi. Describe the differences between primitives and objects
- vii. Demonstrate how an instance of a class is created (new operator)

### (b) Java Programming Basics

- i. Recognize code documentation in source code
- ii. Demonstrate at least two forms of syntax for adding comments to source code
- iii. Replace hard coded constants with named constants

### (c) Standard Java Classes

- i. Demonstrate the use of `JOptionPane.showMessageDialog`
- ii. Demonstrate the use of `JOptionPane.showInputDialog`
- iii. Demonstrate the use of `String.substring`
- iv. Demonstrate the use of `String.length`

### (d) Arithmetic expressions

- i. Demonstrate proper use of the following arithmetic operators: +, -, \*, /,
- ii. Identify and avoid unintended integer division errors

- iii. Distinguish between binary and unary operations
  - iv. Define operator precedence
  - v. Interpret arithmetic expressions following operator precedence rules
  - vi. Define and apply type casting
  - vii. Interpret code that makes use of shorthand assignment operations: `*=`, `/=`, `+=`, `-=`, and `%=`
- (e) Standard input/output
- i. Use wrapper classes to perform type conversion, e.g., `int num = Integer.parseInt("14");`
  - ii. Explain the source of data associated with the system input buffer: `System.in`
  - iii. Perform standard/console input using the `Scanner` class
  - iv. Explain the destination for data sent to the system output buffer: `System.out`
  - v. Perform standard/console output using the `System.out.println` method

### 3. Week 3

- (a) Selection statements
- i. Define the functionality of the following relational operators: `<`, `<=`, `!=`, `==`, `>=`, `>`
  - ii. Use relational operators to control program flow
  - iii. Define the functionality of the following boolean operators: `&&`, `||`, and `!`
  - iv. Use boolean and relational operators to construct meaningful boolean expressions
  - v. Use boolean expressions to control program flow
  - vi. Describe the behavior of an `if` statement
  - vii. Describe the program flow through a series of nested `if` statements
  - viii. Use nested `if` statements to control program flow
  - ix. Use a `switch` statement to control program flow
  - x. Rewrite a `switch` statement with one or more (potentially nested) `if` statements
  - xi. Explain the purpose of the `case`, `break`, and `default` reserved words
- (b) Iteration statements
- i. Interpret code that makes use of the following looping constructs: `while`, `do-while`, and `for`
  - ii. Design and write code that makes use of the following looping constructs: `while`, `do-while`, and `for`
  - iii. Describe how the following constructs differ: `while`, `do-while`, and `for`
  - iv. Rewrite a given `while` and `for` and vice versa

### 4. Week 4

- (a) Midterm exam review
- (b) Midterm Exam 1
- (c) More Standard Java Classes
- i. Define an Application Programming Interface (API)
  - ii. Use Sun's Java documentation to ascertain the capabilities of a given standard Java class
  - iii. Use the Javadoc page for the `Math` class to perform calculations involving the following mathematical operations:
    - A. Absolute value
    - B. Trigonometric functions (in degrees and radians)
    - C. pi-ratio of the circumference of a circle to its diameter
    - D. power functions
    - E. logarithmic functions
    - F. maximum/minimum of two numbers
    - G. Square root

### 5. Week 5

- (a) More Standard Java Classes
- i. Use parsing methods in wrapper classes to convert text representations of numbers into numeric format
  - ii. Use the `toString` method in wrapper classes to convert from numeric format into text representations
  - iii. Be familiar with methods from the `Character` class such as `isDigit` and `toLowerCase`

- iv. Use methods from theStringclass such asisEmpty,substring,indexOf, etc
- v. Generate random numbers
- vi. UseSystem.out.printf to produce formatted output

(b) Java Packages

- i. Explain the purpose of a Java package
- ii. List at least two packages that are part of the Java standard library
- iii. Define the term fully qualified name
- iv. Explain the purpose of theimportstatement

(c) Coding Standards

- i. Explain the purpose of a coding standard
- ii. Be familiar with Sun'scoding conventions.

6. Week 6

(a) Object Oriented Design / Object Oriented Programming

- i. Define the following object oriented concepts:
  - A. Object types (Classes)
  - B. Class instances (Objects)
  - C. Instance variables (Attributes/Fields)
  - D. Instance behaviors/actions (Methods)
- ii. Distinguish between classes and objects
- iii. Describe how objects interact with one another by sending messages
- iv. Demonstrate how to send a message to an object

(b) UML

- i. Correctly annotate and interpret fields (name and type) on a class diagram
- ii. Correctly annotate and interpret methods (with arguments and return type) on a class diagram
- iii. Generating class diagram from a verbal description of a class
- iv. Interpret UML sequence diagrams
- v. Use visibility modifiers to denote the visibility of a field or method

(c) Class creation basics

- i. Define and use classes with multiple methods and data members (fields)
- ii. Define and use value-returning and void methods
- iii. Properly use visibility modifiers in defining methods and fields
- iv. Define and use class constants
- v. Understand and apply accessor and mutator methods
- vi. Distinguish between instance variables and local variables
- vii. Explain what makes a predicate/boolean method unique
- viii. Define and use class methods and instance variables (attributes/fields)
- ix. Define and use methods that have primitive data types as arguments
- x. Understand the importance of information hiding and encapsulation
- xi. Declare and use local variables
- xii. Describe the role of the reserved wordthis
- xiii. Demonstrate use ofthis to disambiguate object fields from local variables

7. Week 7

(a) Defining your own classes

- i. Create and use class constructor methods
- ii. Define and use methods that have reference data types as arguments
- iii. Define and use overloaded methods
- iv. Draw and explain memory diagrams that illustrate the instantiation of objects
- v. Describe the role of the garbage collector
- vi. Compare the equality of two different objects

vii. Avoid redundant code by calling one constructor from a different constructor

8. Week 8

(a) Design Techniques

- i. Use helper methods to avoid redundant code
- ii. Be familiar with various design approaches such as top-down, bottom-up, and case-based

9. Week 9

(a) Class Members

- i. Use class variables/attributes appropriately
- ii. Use class methods appropriately

(b) Arrays

- i. Use an array to store primitive and object types
- ii. Create an array of a given size
- iii. Loop through an array
- iv. Pass an array as an argument

10. Week 10

(a) ArrayLists

- i. Use an *ArrayList*  $\langle E \rangle$  to store objects type  $E$
- ii. Use methods from the *ArrayList*  $\langle E \rangle$  class such as `isEmpty`, `get`, `set`, `add`, `remove`, `size`, `indexOf`, and `lastIndexOf`
- iii. Design and write code that makes use of the enhanced for loop, a.k.a, the for-each loop
- iv. Describe the advantages of an *ArrayList*  $\langle E \rangle$  over an `Array`