

# CSCI 1301

## Exam #1 - Study Guide

### General Information

Date: **February 26th 2009 in class.**

- **The exam will be close book and closed notes.**
  - The exam will consist of practical problems to test your proficiency in the topics taught in class. The level of difficulty of the practical problems will be comparable to those on the labs, pre-lab exercise worksheets, projects and the suggested exercises posted in WebCT./Lectures
- 

The following are topics that may be tested in the 1st Exam.

### Introduction to Computers and Java

**Reading: Java: An Introduction to Problem Solving and Programming Chapter 1 Sections 1.1, 1.2, 1.3 Testing and Debugging and lecture notes.**

- Explain the concepts introduced in class
- Computer System
- Hardware
  - Describe the different hardware components
- Software
  - Program
- Explain what a programming language is.
  - Explain what the Syntax of programming language is.
  - Explain what the Semantic of programming language is.
- Programming Languages
  - Low-level Programming Language
    - Machine language
  - High Level Programming Language

- Differences between low-level programming language and high level programming languages
- Explain what is
  - Assembler
  - Compiler
  - Interpreter
- Explain the steps to translate a Java Program
  - Explain what Java bytecode is.
  - Explain how Java source code is translated and executed.
- Explain the steps in program development
  - Explain what a syntax error is.
  - Explain what a logic error is.
  - Explain what a run-time error is.
- Explain
  - What is an identifier and how to build valid identifiers in Java?
  - What is a reserved word, (also called keyword)?
  - What is a comment and the different types of comments that can be used in java?
- Describe the structure of a Java Application
  - You should be able to write a simple Java application as the ones you have develop for your projects and lab assignments.
    - Declaration of a java class.
    - Declaration of the main method of the class.
- Practice Exercises
  - Self-question in sections **1.1, 1.2, 1.3 Testing and Debugging**
  - Exercises 1, 4
  - Programming Projects 1, 6,9
  - Lab exercises.

## Basic Computation

**Reading: Java: An Introduction to Problem Solving and Programming, Chapter 2 Section 2.1 - 2.4, lecture notes.**

- You should be able to write statements that use the methods **println** and **print** of the object System.out
  - to display string literals on the screen
  - to display string literal that contains escape sequence
  - to display strings and number jointly
- Explain the difference between the method **println** and **print** of the System.out
- Use the string concatenation operator to append one string to another
  - in System.out.println or System.out.print

- Use the string concatenation and string methods to process Strings.
- Explain what a character set is.
- Explain what a Primitive Data type is.
- **Variable**
  - Explain what a variable is.
  - How to declare a variable with and without initialization?
- **Constant**
  - What is a constant?
  - How to declare a constant?
  - Naming Convention for variables and constants
  - Differences between Constants and Variables
- **Assignment statements**
  - Numeric Expressions
    - Order of precedence of arithmetic operators including prefix and postfix mode of the shortcut operators ++,--
    - Evaluation of a numeric expression
    - Assigning the result of a numeric expression to a numeric variable
    - You should be able to compute correctly the value of a numeric expression that contains any number of arithmetic operators including a combination of prefix and postfix shortcut arithmetic operators.
  - How an assignment statement is executed?
  - You should be able to write a shortcut assignment statement and rewrite it as an assignment statement that use the assignment operator =.
- **Data Conversion:**
  - What is a widening conversion?
  - What is a narrowing conversion?
  - You should be able to explain in which situation in a Java Program the type of data conversion is applied:
    - Promotion
    - Assignment conversion
    - Casting
  - You should be able to cast numeric expressions for a data type to a narrower data type
  - You should be able to explain how the casting is applied in a given numeric expression.
- **Interactive Input**
  - How to read input by using objects of the Scanner class
    - Create an object of the class Scanner
    - Use the appropriate methods of a Scanner object to read input.
- **Practice Exercises**
  - Self-question in sections 2.1, 2.2 and 2.4
  - Exercises 1, 2, 3, 4, 6, 11
  - Programming Projects 1, 3, 4, 6, and 9.
  - Lab exercises and worksheets.

# Decision Statements

**Reading: Java: An Introduction to Problem Solving and Programming, Chapter 3**  
**Sections 3.1 - 3.3, lecture notes.**

- What is flow of control of the statements in a program
  - Describe the differences among sequence, decision and repetition statements
- **Boolean Expressions:**
  - Comparisons: Relational Operators.
  - What is lexicographical order?
  - What is the result of using the following methods of a the class String
    - equals
    - equalsIgnoreCase
    - compareTo
  - Logical Operators: AND (&&), OR (||), NOT (!)
    - Truth tables of Logical Operators
    - Shortcut mode of the logical operator AND (&&) and OR (||)
    - Order of Precedence to evaluate Boolean expressions
  - Apply the order of precedence of arithmetic, relational and logical operators and the short-circuit mode of the AND and OR to compute the Truth value of a Boolean expression.
  - Write simple Booleans expression using comparisons.
  - Write complex Boolean expressions involving comparisons of arithmetic expressions and Boolean variables/constants combined by logical operators.
  - Write Boolean expressions from English statements like the exercise worksheet #2.
- **Decision Statements**
  - You should be able to write a Java program that involves if-statement using the appropriate syntax
    - Simple Decision Statements: **if ...**
    - **if ... else** statements
    - Multibranch: **if else if** else statements
    - Nested decisions statements
    - Using if-statements to compare two floating-point values:
      - Explain why it is not appropriate to use the == relational operator to compare the equality of two floating point values.
    - Conditional Operator
      - Syntax
      - Explain how it works.
      - Be able to write assignment statements that use the conditional operator.
  - **Switch Statement**
    - Syntax
    - Type of expressions that can be used in a switch statement

- Default case
    - How it works
  - Explain how a switch statement is executed
  - Explain how a break statement works within a switch statement
- You should be able to use decision statements to write Java code that solves problems as the ones proposed in the labs, projects as well as the examples in the textbook.
- You should be able to interpret and understand code that uses decision statements (**if, if-else, if-else-if, switch**) to compute and display some values.
- **Practice Exercises**
  - Self-question in sections 3.1 - 3.3
  - Evaluating Boolean Expression exercises (WebCT/Lecture Notes/Lecture 3)
  - Exercises 1, 2, 3, 4, 5, 6, 11, 12, 13, 15 and 16.
  - Programming Projects: 1, 2, 5, 6, 8 and 9.
  - Lab exercises and pre-lab worksheet.

## Repetition Statements

**Reading: Java: An Introduction to Problem Solving and Programming, Chapter 4  
Sections 4.1 - 4-2 (except nested loops): while, do-while and for loops and lecture notes.**

- **Do-while and while Loops:**
  - Boolean condition as control mechanism of a loop.
  - **While loops:**
    - Syntax
    - Explain how a while statements executes.
  - **Do while Loops:**
    - Syntax
    - Explain how a do while is executed.
  - Indicate the differences between do-while and while statements.
  - Explain what is a sentinel value
- **For Loops**
  - Syntax of a For Loop.
  - Identify and explain the characteristics of a for loop.
  - Explain how a for loop is executed.
  - Be able to write a for-loop as a while loop
- **Loop Bugs**
  - **Infinite Loop**
    - Explain what an infinite loop is.
    - should be able identify when an infinite loop occurs in a Java code
- **Using loops**
  - Solve simple problems using for, while and do-while loops
  - Solve a given problem using the appropriate combination of loops
- You should be able to :
  - Determine the number of iterations executed by a given **while** loop.

- Determine the number of iterations executed by a given **do-while** loop.
- Determine the number of iterations executed by a given **for** loop.
- Determine the output of execution of a loop.
- **Practice Exercises**
  - Self-questions Chapter 4: section 4.1 - 4.2 (except nested loops).
  - Exercises 1, 2, 4, 5, 8, 9, 13, 18 and 19.
  - Programming Projects, 3, 6, 7, 9, 13.
  - Loop Exercise Worksheet - WebCT/Lecture/Loops.