Computer Programming I

COP 2210

Syllabus

Spring Semester 2012

Instructor: Greg Shaw

Office: ECS 313 (Engineering and Computer Science Bldg)

Office Hours: Tuesday: 2:50 – 4:50, 7:45 – 8:30
Thursday: 2:50 – 4:50, 7:45 – 8:30
(all times pm, others by appointment)

Phone: (305) 348-1550

E-mail: shawg@fiu.edu

Web: http://www.cs.fiu.edu/~shawg

Required Text: Big Java
4th Edition
by Cay Horstmann

FIU COP 2210 Common Course Objectives

1. Be familiar with the concepts of Objects and Classes
2. Master using the fundamental Java data types
3. Master using the Java selection and iteration constructs
4. Master using String and ArrayList classes
5. Master analyzing problems and writing Java program solutions to those problems using the above features

1.) Introduction - *Chapter 1*

A Brief History of Computer Languages
Machine Languages, Assembly Languages, High-Level Languages
Translating Human-Readable Programs to Machine Language
Compiled Languages, Interpreted Languages, and Java
String Literals ("Constants") and String Concatenation
Escape Sequences
The ASCII and Unicode Character Sets
Errors
Syntax Errors, *Exceptions (aka: Run-time Errors)*, and *Logic Errors* (i.e., *Semantic* Errors)
Algorithms and Problem-Solving
Using the NetBeans IDE

2.) Using an Existing Class (i.e. Creating and Manipulating Objects) - *Chapter 2*

Introduction to Variables and Data Types
The Assignment Statement
Object-Oriented Programming (OOP) Concepts
   Classes, Objects, and Methods
Introduction to the String Class
   Creating String objects, and the *length*, *replace*, *indexOf*, and *substring* methods
Constructing ("Creating") Objects
Objects, Object Variables, and Object References
Assignment of Object Variables
Using Objects (i.e. Calling Methods for Objects)
Accessor and Mutator Methods (aka: "get" and "set" Methods)
Methods That Return a Value vs. "void" Methods
Local Variables

3.) Implementing ("Creating") Classes - *Chapter 3*

Class *Interface* vs. Class *Implementation*
   Encapsulation and Information Hiding
Defining Classes and Methods
Instance Variables (aka: Instance *Fields*)
   Access Specifier, Type, and Name
Class Constructors
Parameter Variables (aka: Method *Parameters*)
Variable Scope, Lifetimes and Initial Values
The **this** Object Reference and Shadowing
Method Overloading

4.) Primitive Data Types (and More) - *Chapter 4*

Java's Primitive Data Types: **int**, **double**, **char**, **boolean**
Arithmetic Operators and Operator Precedence
Integer Arithmetic and Mixed-Type Arithmetic
   Integer Division and the Modulus ("Mod") Operator
Type Conversion (aka: Type *Casting*) and "Roundoff" Errors
The "Shortcut" or "Arithmetic" Assignment Operators
The Increment and Decrement Operators
Defined Constants (i.e., **final** variables)
Intro to **static** Methods
Math Class Methods (i.e., "Functions") - see pgs. 141, 1002-1004
Reading User Input
   Using the *showInputDialog* method of the *JOptionPane* Class
Explicit vs. Implicit Method Parameters
Assignment of Primitive Types and Assignment of Objects
   The Meaning of "="
   Object References and Aliases

5.) Style and Documentation Standards for Java Programs (*Online Notes and Appendices H and L*)

Style Considerations – Creating "Readable" Programs
Java "Documentation Comments" (aka: "javadoc" Comments)
"Internal" Documentation

6.) Decision-Making (aka: Selection, Conditional Execution) – *Chapter 5*

Relational Operators and Relational Expressions
The **if** Statement
   Single-Alternative Decisions ("*yes/no*")
   Two-Alternative Decisions ("*either/or*")
"Nested" **if** Statements
   Forming More Complex Conditions
   Multiple-Alternative Decisions ("*one of many*")
Testing "Equality" of Floating-Point Numbers
String Comparisons
   The *equals* Method vs. the Equality Operator ("==")
   The *equalsIgnoreCase* Method
Type **boolean**
   boolean Operators and Evaluating boolean Expressions
   boolean Variables ("flags") and the boolean Assignment Statement
   boolean Methods (aka: "Predicate" Methods)
   "Lazy" (or, "Shortcut") Evaluation of boolean Expressions
DeMorgan's Laws for Simplifying Boolean Expressions
Decision-Making Pitfalls
   Testing Programs that make Decisions
   Impossible Conditions and *Unavoidable* Conditions
   The "Dangling Else" Problem (How to Avoid It)

7.) Iteration (aka: Repetition, Looping) - *Chapter 6*

The **while** Loop
Loop Necessities
Defensive Programming and "Robust" Programs
   Using **while** to Validate Input
Accumulators and *Counters*
The **for** Loop
The **do** Loop (aka: The "*do-while*" Loop)
Reading Data Until End-of-File
Introduction to the Scanner class
   Methods *next*, *nextInt*, and *nextDouble*, and boolean Method *hasNext*
The "Loop and a Half" Problem and the **break** Statement
Nested Loops
Iteration Pitfalls: Infinite Loops and "Off by One" Errors

8.) The *ArrayList* Class – *Chapter 7, Sections 7.2 and 7.3 ONLY!*

"Generic" ArrayLists
ArrayList Methods *add*, *get*, *size*, *remove*, *set*, and *clear*
ArrayLists of Primitive Types
   "Wrapper" Classes, Autoboxing, and Autounboxing
ArrayLists of Objects

9.) Files – *Online Notes and Chapter 11, Sections 11.1 and 11.2*

File Concepts
   Sequential Access vs. Random Access Files
   ASCII Files vs. Binary Files
Reading from Input Files ("Data Files") Using the Scanner Class
Writing to Output Files Using the PrintWriter Class

10.) Object-Oriented Design - *Chapter 8*

Choosing Classes to Model
Class Cohesion, Class Coupling, and Method "Side Effects"
Call-by-Value vs. *Call-by-Reference* Parameter Passing
Why you can't change the value of a method *argument*
Proper Method Documentation: *Preconditions* and *Postconditions*
Static Class Methods and Static Class Variables

11.) The String Class Revisited – *Online Notes and Chapter 4, Section 4.5*

The *null* ("empty") String vs. the **null** Object *Reference*
String Class Methods *length*, *substring*, *indexOf*, *toUpperCase*, *toLowerCase*, **and** *charAt*
String Comparisons - the *compareTo* and *compareToIgnoreCase* Methods

---

Assigned Rooms (check your schedule for exact days and times)

Class: ECS 135

Labs (*required*): ECS 141

Tutoring (*optional*): ECS 235

☞ See "*Understanding Your Schedule*," online

☞ The John C. Comfort Undergraduate Lab is ECS 241. This is an "open" lab where you can work at any time

Important Dates

Midterm Exam – Thursday, February 23rd

Drop Date - Friday, March 2nd

Final Exam

Sections U1 and U2 - Thursday, April 26th (9:45 – 11:45 am)

Sections U3 and U4 - Tuesday, April 24th (12:00 – 2:00 pm)

Sections U5 and U6 - Thursday, April 26th (5:00 – 7:00 pm)

Composition of Course Grade

Item	Value
Programming assignments (8 or 9)	25%
Midterm exam	25%
Final exam	40%
Lab assignments	10%

☞ ***You must pass the tests to pass the class. I.e. the average of your two test scores must be at least 60% of the highest average in the class.***

Computation of Final Course Grade

1. Final numeric averages are "curved" by comparing each student's numeric average (see above) to the highest in the class. For example, suppose the highest average in the class is 90%. Then, an average of 75 would "curve" to an 83, because 75 is 83% of 90.

2. The curved numeric average is then converted to a letter grade according to this scale:

Numeric Average	Letter Grade
93..100	A
90..92	A-
87..89	B+
83..86	B
80..82	B-
77..79	C+
70..76	C
60..69	D
0..59	F

Other Important Information

Class policies on late assignments, partial credit, makeup tests, academic honesty, incompletes, etc, are covered in the online document "Class Rules and Gregulations."