

## School of Computing and Information Sciences

**Course Title:** Computer Programming I - Lab

**Date:** 2/7/2020

**Course Number:** COP 2210L

**Number of Credits:** 0

<b>Subject Area:</b> Programming	<b>Subject Area Coordinator:</b> Janki Bhimani <b>email:</b> jbhimani@fiu.edu
<b>Catalog Description:</b> Lab component for Computer Programming I Lecture.	
<b>Textbook:</b> Big Java by Cay Horstmann	
<b>References:</b>	
<b>Prerequisites Courses:</b> None	
<b>Corequisite Courses:</b> COP 2210	

**Type:** Required Common Prerequisite

**Prerequisites Topics:**

- Mathematical functions
- Arithmetic and geometric sequences

**Lab Outcomes:**

- O1. Be familiar with the concepts of Objects & Classes
- O2. Master the fundamental Java data types
- O3. Master the Java selection and iteration constructs
- O4. Be familiar with arrays & ArrayLists
- O5. Master using String and Wrapper classes
- O6. Be familiar with reading and writing of text files
- O7. Master analyzing problems and writing Java program solutions to those problems using the above features
- O8. Be exposed to software testing and interactive debugging
- O9. Master complex Boolean expressions in selection and iteration constructs
- O10. Master good programming practices
- O11. Master methods, method parameters, and parameter passing

*(Subject Area Coordinator will provide a list of best programming practices for instructors as a reference)*

**Relationship between Lab Outcomes and Program Outcomes**

<b>BS in CS: Program Outcomes</b>	<b>Course Outcomes</b>
a) Demonstrate proficiency in the foundation areas of Computer Science including mathematics, discrete structures, logic and the theory of algorithms	
b) Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems.	1, 2, 3, 4, 5
c) Demonstrate proficiency in problem solving and application of software engineering techniques	1, 2, 3, 4, 5
d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.	1, 2, 3, 4, 5
e) Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.	
f) Demonstrate the ability to work cooperatively in teams.	
g) Demonstrate effective communication skills.	

**Assessment Plan for the Course & how Data in the Course are used to assess Program Outcomes**

Student and Instructor Course Outcome Surveys are administered at the conclusion of each offering, and are evaluated as described in the School's Assessment Plan:  
<http://www.cis.fiu.edu/programs/undergrad/cs/assessment/>

School of Computing and Information Sciences

COP 2210L

Programming I - Lab

**Course Description:** The lab for a first course in computer science that uses a structured programming language to study programming and problem solving on the computer. Includes the design, construction and analysis of programs. Student participation in a closed instructional lab is required.

**Outline**

<b>Topic</b>	<b>Number of Lecture Hours</b>	<b>Outcome</b>
<ul style="list-style-type: none"><li>● <b>Object-Oriented Design and Programming</b></li><li>○ Classes</li><li>○ Objects</li><li>○ Methods</li></ul>	<b>10</b>	<b>O1, O5</b>
<ul style="list-style-type: none"><li>● <b>I/O</b></li><li>○ JOptionPane</li><li>○ Text files</li></ul>	<b>5</b>	<b>O1, O5</b>
<ul style="list-style-type: none"><li>● <b>Fundamental Java data types</b></li><li>○ Primitive types</li><li>○ Strings</li><li>○ Wrapper classes</li></ul>	<b>5</b>	<b>O2, O5</b>
<ul style="list-style-type: none"><li>● <b>Control structures</b></li><li>○ Selection</li><li>○ Iteration</li><li>○ Logic</li></ul>	<b>10</b>	<b>O3, O5</b>
<ul style="list-style-type: none"><li>● <b>Arrays and ArrayLists</b></li></ul>	<b>6</b>	<b>O4, O5</b>

## **Learning Outcomes (Familiarity---> Usage ---> Assessment)**

### **Object-Oriented Design and Programming**

1. Understand Classes, access modifiers, and encapsulation [Familiarity]
2. Apply UML (Unified Modeling Language) class diagrams to design classes [Familiarity]
3. Create Class Constructors [Usage]
4. Create and use Methods with multiple arguments [Usage]
5. Apply Instance fields and methods [Usage]
6. Understand Object instantiation and variable scope [Familiarity]
7. Understand and apply of **this** and **null** references [Usage]
8. Understand and apply of **toString** and **equals** method [Usage]
9. Utilize methods from the Math and Random class [Usage]

### **I/O**

1. Write data to text files [Usage]
2. Append data to existing text files [Usage]
3. Read data from existing text files, and detect end of file [Usage]
4. Request input from the user using Scanner class [Usage]
5. Request input from the user using JOptionPane class [Usage]
6. Output data to the console using System.out [Usage]
7. Output data to the console using JOptionPane class [Usage]

### **Fundamental Java Data Types**

1. Understand the different data types, their ranges, and their uses [Familiarity]
2. Master variable declarations and assignments using the appropriate Java data types [Usage]
3. Use String literals and escape sequences appropriately [Usage].
4. Create expressions & statements using the correct data types and logic [Implementation].
5. Understand the application of arithmetic operators, increment/decrement operators, integer division, precedence, real numbers, and mixing types [Usage].
6. Implement naming conventions for variable declarations [Implementation].
7. Analyze the scope of a variable, and implement it according to the logic of the program. [Assessment]
8. Master the use of casting variables to other data types when needed. [Assessment]

### **Control Structures**

1. Understand selections and conditions using Boolean variables and expressions [Familiarity]

School of Computing and Information Sciences  
COP 2210L  
Programming I - Lab

2. Master the use of if, if-else statements, and nested if's. [Usage]
3. Understand the use of switch statements. [Familiarity]
4. Analyze when to use if-statements vs switch statements. [Assessment]
5. Master the use of compound conditions using logical operators and their precedence. [Implementation]
6. Understand how to test the value of Strings and other objects. [Usage]
7. Understand how to test the value of primitives. [Usage]
8. Decide the appropriate testing expressions according to the logic. [Assessment]
9. Master the syntax of the do-while, while, and for loops, and their nesting. [Usage]
10. Evaluate when to use each type of loop, and effectively control their iterations. [Assessment]

### **Arrays & ArrayLists**

1. Master the declaration, initialization, and access of arrays, for primitives and objects. [Usage]
2. Determine the length of arrays, and how to prevent off-by-one errors. [Assessment]
3. Understand the use of enhanced for loops in arrays, and its limitations. [Assessment]
4. Master the ability to pass arrays as arguments, compare arrays, and copy arrays. [Implementation]
5. Understand how to use arrays to store and retrieve data in text files. [Implementation]

**Course Outcomes Emphasized in Laboratory Projects & Assignments**

Outcome	Number of Weeks
	<b>Approximately 24 hours of assignments are given. In addition, students complete approximately 12 hours of lab instruction.</b>
<b>O1, O7, O10, O11</b>	Assignment(s) with the concepts of Objects, Classes, Methods & parameters (10 hours)
<b>O2, O7, O10</b>	Assignment(s) with the fundamental Java data types (2 hours)
<b>O3, O7, O8, O9, O10</b>	Assignment(s) with Java selection and iteration constructs (10 hours)
<b>O4, O7, O8, O10</b>	Assignment(s) with Arrays & ArrayLists (6 hours).
<b>O5, O7, O10</b>	Assignment(s) with String and Wrapper Classes (3 hours)
<b>O6, O7, O10</b>	Assignment(s) with JOptionPane, reading and writing of text files ( 5 hours)

**Oral and Written Communication:**

None

**Social and Ethical Implications of Computing Topics:**

None

**Approximate number of credit hours devoted to fundamental CS topics**

Topic	Core Hours	Advanced Hours
<b>Algorithms:</b>	<b>1.0</b>	
<b>Software Design:</b>	<b>1.0</b>	
<b>Computer Organization and Architecture:</b>	<b>0</b>	
<b>Data Structures:</b>	<b>0</b>	

School of Computing and Information Sciences  
COP 2210L  
Programming I - Lab

<b>Concepts of Programming Languages:</b>	<b>1.0</b>	
---	------------	--

**Theoretical Contents:**  
**None**

**Problem Analysis Experiences:**  
**None**

**Solution Design Experiences**

<b>Approximately 24 hours of assignments are given..</b>
<b>In addition, students complete approximately 12 hours of lab instruction.</b>

**The Coverage of Knowledge Units within Computer Science Body of Knowledge<sup>1</sup>**

<b>Knowledge Unit</b>	<b>Topic</b>	<b>Lecture Hours</b>
<b>DS 2</b>	<b>Logic (and/or, etc)</b>	<b>2</b>
<b>PL 1</b>	<b>Control structures (Selection &amp; Iteration)</b>	<b>8</b>
<b>PL 3</b>	<b>Arrays, ArrayLists</b>	<b>6</b>
<b>PL 5</b>	<b>I/O with JOptionPane, Text Files</b>	<b>5</b>
<b>PL 4</b>	<b>Primitives, Strings, and Wrapper classes</b>	<b>5</b>
<b>PL 6</b>	<b>Objects, Classes, &amp; Methods</b>	<b>10</b>

---

<sup>1</sup>

See [https://www.acm.org/binaries/content/assets/education/cs2013\\_web\\_final.pdf](https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf) for a description of Computer Science Knowledge units.