

## School of Computing and Information Sciences

**Course Title:** Computer Programming II

**Date:** 2/12/2018

**Course Number:** COP 3337

**Number of Credits:** 3

<b>Subject Area:</b> Programming	<b>Subject Area Coordinator:</b> Tim Downey <b>email:</b> downeyt@cs.fiu.edu
<b>Catalog Description:</b> An intermediate level course in Object-Oriented programming. Topics include primitive types, control structures, strings, arrays, objects and classes, data abstraction inheritance polymorphism and an introduction to data structures. This course will have additional fees.	
<b>Textbook:</b> Big Java by Cay Horstmann	
<b>References:</b>	
<b>Prerequisite Courses:</b> (COP2210 or EEL2880)	
<b>Co-requisite Courses:</b> None	

**Type:** Required

### **Prerequisites Topics:**

- Be familiar with Objects & Classes
- Master methods, method parameters, and parameter passing
- Master fundamental Java data types
- Master selection and iteration control structures
- Master using String, ArrayList, and Wrapper classes
- Be exposed to software testing and interactive debugging
- Master complex Boolean expressions in selection and iteration constructs
- Master good programming practices

### **Course Outcomes:**

- O1. Master the design and implementation of classes using inheritance and polymorphism**
- O2. Master the use and implementation of class interfaces**
- O3. Be familiar with writing recursive methods**
- O4. Be familiar with the implementation of linked list data structures**
- O5. Be familiar with the Stack & Queue data structures**
- O6. Be exposed to the Java Collection interface**
- O7. Master analyzing problems and writing Java program solutions to those problems**
- O8. Be familiar with software testing and interactive debugging**
- O9. Master best practices for documenting code**
- O10. Master arrays and multidimensional arrays**

*(SAC will provide a list of best programming practices for instructors as a reference)*

**School of Computing and Information Sciences**  
**COP 3337**  
**Programming II**

**Relationship between Course 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	3, 5 ,7
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, 6, 7
c) Demonstrate proficiency in problem solving and application of software engineering techniques	1, 2, 3, 4, 5, 6, 7
d) Demonstrate mastery of at least one modern programming language and proficiency in at least one other.	1, 2, 3, 4, 5, 6, 7
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 3337**  
**Programming II**

**Outline**

<b>Topic</b>	<b>Number of Lecture Hours</b>	<b>Outcome</b>
<ul style="list-style-type: none"> <li>• <b>Review of Java Basics</b> <ul style="list-style-type: none"> <li>○ Primitive types</li> <li>○ Class types</li> <li>○ Static objects and methods</li> <li>○ Control structures</li> <li>○ String, ArrayList &amp; Wrapper Classes</li> </ul> </li> </ul>	<b>6</b>	
<ul style="list-style-type: none"> <li>• <b>Text I/O</b> <ul style="list-style-type: none"> <li>○ BufferedReader</li> <li>○ FileReader</li> </ul> </li> </ul>	<b>1</b>	
<ul style="list-style-type: none"> <li>• <b>Introduction to Inheritance</b> <ul style="list-style-type: none"> <li>○ Object class</li> </ul> </li> </ul>	<b>2</b>	<b>O1 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Data Abstraction</b> <ul style="list-style-type: none"> <li>○ Implement data type classes</li> <li>○ Constructors</li> <li>○ Override equals, hashCode and toString</li> </ul> </li> </ul>	<b>5</b>	<b>O1 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Interfaces</b> <ul style="list-style-type: none"> <li>○ Implementing the Comparable interface</li> <li>○ Implementing the Comparator interface</li> <li>○ Collection class</li> </ul> </li> </ul>	<b>3</b>	<b>O2 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Inheritance &amp; Polymorphism</b> <ul style="list-style-type: none"> <li>○ Extending classes</li> <li>○ Overloading and overriding methods</li> <li>○ Polymorphism</li> </ul> </li> </ul>	<b>7</b>	<b>O1, O2 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Exceptions</b></li> </ul>	<b>1</b>	<b>O1 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Recursion</b> <ul style="list-style-type: none"> <li>○ Stack Frames</li> </ul> </li> </ul>	<b>4</b>	<b>O3 &amp; O7</b>
<ul style="list-style-type: none"> <li>• <b>Introduction to Data Structures</b> <ul style="list-style-type: none"> <li>○ Collection interface</li> <li>○ Stacks</li> <li>○ Queues</li> <li>○ Implementing linked lists</li> </ul> </li> </ul>	<b>9</b>	<b>O4,O5,O6 &amp; O7</b>

**School of Computing and Information Sciences  
COP 3337  
Programming II**

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

<b>Outcome</b>	<b>Number of Weeks</b>
<b>O1</b>	<b>2 assignments 4 weeks</b>
<b>O2</b>	<b>1 assignment 2 weeks</b>
<b>O3</b>	<b>1 assignment 2 weeks</b>
<b>O4,O5,O6</b>	<b>1-2 assignments 2-4 weeks</b>

**Oral and Written Communication:  
None**

**Social and Ethical Implications of Computing Topics:  
None**

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

<b>Topic</b>	<b>Core Hours</b>	<b>Advanced Hours</b>
<b>Algorithms:</b>	<b>0.5</b>	
<b>Software Design:</b>	<b>1.0</b>	
<b>Computer Organization and Architecture:</b>	<b>0</b>	
<b>Data Structures:</b>	<b>0.5</b>	
<b>Concepts of Programming Languages:</b>	<b>1.0</b>	

**School of Computing and Information Sciences**  
**COP 3337**  
**Programming II**

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

**Problem Analysis Experiences**

**6 assignments**

**Solution Design Experiences**

**6 assignments**

**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>PF1</b>	<b>Text I/O</b>	<b>1</b>
<b>PF3</b>	<b>Introduction to Data Structures</b>	<b>9</b>
<b>PF4</b>	<b>Recursion</b>	<b>4</b>
<b>PF5</b>	<b>Event Driven Programming</b>	<b>1</b>
<b>PL6</b>	<b>Object Oriented Programming</b>	<b>14</b>
<b>SE2</b>	<b>Using APIs</b>	<b>1</b>

<sup>1</sup>See <http://www.computer.org/education/cc2001/final/chapter05.htm> for a description of Computer Science Knowledge units