

## School of Computing and Information Sciences

**Course Title:** Intermediate Java Programming

**Date:** 2/12/2018

**Course Number:** COP 3804

**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> A second course in Java programming. Continues Programming in Java by discussing object-oriented programming in a more detail, with larger programming projects and emphasis on inheritance. Not acceptable for credit for CS majors. This course will have additional fees.	
<b>Textbook:</b> Starting Out with Java 5: Control Structures to Objects; Gaddis	
<b>References:</b> None	
<b>Prerequisites Courses:</b> (COP2250 or COP2210)	
<b>Co-requisites Courses:</b> None	

Type: Required

Prerequisites Topics:

- Java data types
- Java fundamentals including String, ArrayList, classes
- Problem solving using java

Course Outcomes:

1. Master the design and implementation of classes using inheritance and polymorphism.
2. Master the use and implementation of Class interfaces
- ~~3. Master the uses of one- and two-dimensional arrays~~
- ~~4. Be familiar with software testing methods and interactive debugging~~
- ~~5. Be familiar with standard Java collection classes~~
- ~~3. Be exposed to writing recursive methods~~
- ~~4. Be exposed to the implementation of linked list structures~~
- ~~5.6.~~ Master analyzing problems and writing java program solutions to those problems using the above features

**School of Computing and Information Sciences**  
**COP 3804**  
**Intermediate Java Programming**

**Outline**

<b>Topic</b>	<b>Number of Lecture Hours</b>	<b>Outcome</b>
<ul style="list-style-type: none"> <li>• Java preliminaries               <ul style="list-style-type: none"> <li>○ Java types, classes and objects</li> <li>○ Class variables, Final variables, etc.</li> <li>○ Static methods and objects</li> <li>○ Input/Output (using Files, too)</li> <li>○ String class – Wrapper classes</li> <li>○ ArrayList class</li> </ul> </li> </ul>	6	
<ul style="list-style-type: none"> <li>• Classes and Interfaces               <ul style="list-style-type: none"> <li>○ Data abstraction</li> <li>○ Class constructors</li> <li>○ Interfaces</li> <li>○ Overriding equals and toString methods</li> <li>○ Comparable and Comparator interfaces</li> <li>○ Public, Private, and Protected access specifiers</li> <li>○ Overloading method names</li> </ul> </li> </ul>	11	2,5
<ul style="list-style-type: none"> <li>• Inheritance &amp; Polymorphism               <ul style="list-style-type: none"> <li>○ Introduction to Inheritance – Reuse of code</li> <li>○ Overriding method names</li> <li>○ Polymorphism and its applications</li> </ul> </li> </ul>	7	1,5
<ul style="list-style-type: none"> <li>• Recursion               <ul style="list-style-type: none"> <li>○ Definition</li> <li>○ Why does it work</li> <li>○ Infinite recursion</li> <li>○ Examples</li> </ul> </li> </ul>	4	3,5
<ul style="list-style-type: none"> <li>• Introduction to Data Structures               <ul style="list-style-type: none"> <li>○ Linked List and its implementation</li> <li>○ Stack &amp; Queue data structures</li> <li>○ Introduction to Java Collection Interface</li> </ul> </li> </ul>	8	4,5

**School of Computing and Information Sciences**  
**COP 3804**  
**Intermediate Java Programming**

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

	<b>Outcome</b>	<b>Number of Weeks</b>
1	Classes and Objects Outcome: 5	2
2	Inheritance & Polymorphism Outcomes: 1,5	3
3	Interface implementation Outcomes: 2,5	3
4	Recursive methods Outcomes: 3,5	3
5	Data Structures Outcomes: 4,5	3

**Oral and Written Communication:** No significant coverage

Number of written reports:

Approximate number of pages for each report:

Number of required oral presentations:

Approximate time for each presentation:

**Social and Ethical Implications of Computing Topics**

No significant coverage

<b>Topic</b>	<b>Class time</b>	<b>Student performance measures</b>