

Knight Foundation School of Computing and Information Sciences

Course Title: Computer Programming II

Date: 2/12/2018

Course Number: COP 3337

Number of Credits: 3

Subject Area: Programming	Subject Area Coordinator: Janki Bhimani email: jbhimani@fiu.edu
Catalog Description: 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.	
Textbook: Big Java by Cay Horstmann	
References:	
Prerequisite Courses: (COP2210 or EEL2880)	
Co-requisite Courses: 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**

Knight Foundation School of Computing and Information Sciences
COP 3337
Programming II

Relationship between Course Outcomes and Program Outcomes

BS in CS: Program Outcomes	Course Outcomes
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:
<https://abet.cs.fiu.edu/csassessment/>

Knight Foundation School of Computing and Information Sciences
COP 3337
Programming II

Outline

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

Knight Foundation School of Computing and Information Sciences
COP 3337
Programming II

Course Outcomes Emphasized in Laboratory Projects / Assignments

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

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
Algorithms:	0.5	
Software Design:	1.0	
Computer Organization and Architecture:	0	
Data Structures:	0.5	
Concepts of Programming Languages:	1.0	

Theoretical Contents:
None

Problem Analysis Experiences

6 assignments

Solution Design Experiences

6 assignments

Knight Foundation School of Computing and Information Sciences
COP 3337
Programming II

The Coverage of Knowledge Units within Computer Science Body of Knowledge¹

Knowledge Unit	Topic	Lecture Hours
PF1	Text I/O	1
PF3	Introduction to Data Structures	9
PF4	Recursion	4
PF5	Event Driven Programming	1
PL6	Object Oriented Programming	14
SE2	Using APIs	1

¹See https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf for a description of Computer Science Knowledge units