

**Knight Foundation School of Computing and Information Sciences  
COP 3XXX Python Programming II**

**Knight Foundation School of Computing and Information Sciences**

**Course Title:** Python Programming II

**Date:** 10/16/2023

**Course Number:** COP 3XXX

**Number of Credits:** 3

<b>Subject Area:</b> Programming	<b>Subject Area Coordinator:</b> Janki Bhimani <b>email:</b> jbhimani@fiu.edu
<b>Catalog Description:</b> Object-oriented principles, handling modules, packages, and decorators, working with databases, data structures, and visualization tools. More complex Pythonic solutions for real-world challenges.	
<b>Textbooks:</b> Python Object-Oriented Programming: Build robust and maintainable object-oriented Python applications and libraries by Steven F. Lott, Dusty Phillips, 4th Edition. ISBN-13: 978-1801077262. Publisher: Packt Publishing. Date: July 2, 2021.	
<b>References (for further reading):</b>	
<b>Prerequisites Courses:</b> COP 2XXX - Python Programming I or Advisor's Permission	
<b>Corequisite Courses:</b> COP 2XXX - Introduction to Data Science and COP 3XXX - Computational Thinking	

Type: Core Course for BS in Data Science; Elective for CS and IT Majors.

Prerequisites Topics:

1. Programming fundamentals such as control structures, basic data types and structures, functions, and object-oriented paradigm in at least one programming language.
2. Experience in solving simple computational problems using coding.

Course Outcomes:

1. **Design, implement, and debug** object-oriented applications utilizing principles like objects, interfaces, inheritance, and polymorphism.
2. **Make use of** Python's module system, creating, importing, and distributing packages to enhance code reusability and manage large projects.
3. **Build** Python applications that can connect to, query, and manipulate databases, and other web services.
4. **Apply** complex algorithms and data structures to solve real-world problems.
5. **Utilize** Python libraries like pandas for data processing tasks.
6. **Visualize** data using Python's matplotlib library, producing graphs and charts.
7. **Compare** advanced programming constructs like decorators, context managers, and iterators.
8. **Identify** recursion in order to enhance problem-solving skills for both computational challenges.
9. **Examine and differentiate** the principles and applications of data structures, understanding their significance in algorithm design and optimization.

**Knight Foundation School of Computing and Information Sciences  
COP 3XXX Python Programming II**

**Association between Student Outcomes and Course Outcomes**

<u><b>BS in Computing: Student Outcomes</b></u>	<b>Course Outcomes</b>
1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	1,2,3,4,5,6,7,8,9
2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.	1,2,3,4,5,6,7,8,9
3) Communicate effectively in a variety of professional contexts.	
4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.	
5) Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline.	
<u><b>Program Specific Student Outcomes</b></u>	
6) Apply theory, techniques, and tools throughout the data science lifecycle and employ the resulting knowledge to satisfy stakeholders' needs. [DS]	4,5,6,9

**Assessment Plan for the Course and how Data in the Course are used to assess Student 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.cis.fiu.edu/>

**Knight Foundation School of Computing and Information Sciences**  
**COP 3XXX Python Programming II**

**Outline**

<b>Topic</b>	<b>Number of Lecture Hours (Total: 37.5 hours = 15 weeks * 2 lectures/week * 1.25 hrs/lecture)</b>	<b>Outcome</b>
1. <u>Objects in Python</u> 1.1. Init method 1.2. Constructors	3	1
2. <u>Object-oriented programming II</u> 2.1. Encapsulation and class methods 2.2. Inheritance and polymorphism	4.5	1
3. <u>Advanced Topics in Object-oriented programming II</u> 3.1. Abstract classes (ABC) and Interfaces 3.2. Method overloading	3.75	1
4. <u>Modules and packages</u> 4.1. Importing and using modules 4.2. Creating and distributing packages	3.75	7
5. <u>Iterators and generators</u> 5.1. Iterators and the iterator protocol 5.2. Generators and generator expressions	3	7
6. <u>Decorators and context managers</u> 6.1. Introduction to decorators and their usage 6.2. Creating and using context managers	3	3,4
7. <u>Recursion, sorting, and searching in Python</u> 7.1. Introductory topics in recursion, sorting, and searching algorithms	4.5	8
8. <u>Data Structures</u> 8.1. Introduction to stacks, queues, trees	3.75	4,9
9. <u>Working with databases</u> 9.1. Introduction to databases and SQL 9.2. Python and SQLite	4.5	3,4,5,6
10. <u>Data processing and visualization</u> 10.1. Introduction to data processing with pandas 10.2. Data visualization with matplotlib	3.75	6,9

**Performance Measures for Evaluation**

All assignments are assigned through the Canvas course site. The deadlines are strictly enforced. For example, if the deadline is 11:59 PM, any assignment submitted after this time is considered

**Knight Foundation School of Computing and Information Sciences**  
**COP 3XXX Python Programming II**

late. It is also each student's responsibility to submit correct files and ensure the submission is successful before the deadline. If students are unable to submit their assignment through Canvas, they will need to send a copy of their assignment to the instructor before the stated deadline. There will be three exams and each exam will be cumulative with an emphasis on the most recently covered material. Exam details will be posted on the Canvas course site (<https://canvas.fiu.edu>).

Assignment	Total Points	Percentage of Final Grade
Quizzes (11-Drop-1)	100 each	10%
Homework Assignments (3)	100 each	30%
Exam 1	100	20%
Exam 2	100	20%
Exam 3	100	20%
<b>TOTAL</b>		100%

**Letter Grade Distribution Table**

Letter	Range%	Letter	Range%	Letter	Range%
A	93 or above	B	82 - 85.9	C	70 - 73.9
A-	90 - 92.9	B-	78 - 81.9	D	60 - 69.9
B+	86 - 89.9	C+	74 - 77.9	F	less than 60

**Description of Possible Homework Activities**

**Homework 1: Object-Oriented Programming (OOP)**

Description: Students are required to design a basic library management system using object-oriented programming concepts. The system should allow adding books, checking out books, and displaying all books. Implement classes for books and users, using encapsulation, inheritance, and polymorphism.

**Rubric:**

<b>Implementation of Classes (30 points)</b>	'Book' class with attributes like title, author, ISBN	15 points
	'User' class inheriting from 'Book' with methods to check out and return books	15 points
<b>Application of Inheritance (20 points)</b>	Effective inheritance structure and use of polymorphism in methods.	20 points
<b>System Functionality (30 points)</b>	System can add, check out, and display books	30 points
<b>Code Quality (20 points)</b>	Clean, commented, and organized code.	20 points

**Knight Foundation School of Computing and Information Sciences**  
**COP 3XXX Python Programming II**

**Homework 2: Modules and Data Structures**

Description: Students need to implement a module that offers operations on stacks and queues. Furthermore, they'll write a test script importing this module, showcasing all its functionalities.

**Rubric:**

<b>Module Creation (20 points)</b>	Properly defined module for stacks and queues	20 points
<b>Stack Operations (40 points)</b>	Push, Pop, and Display operations	20 points
<b>Queue Operations (20 points)</b>	Enqueue, Dequeue, and Display operations	20 points
<b>Test Script (20 points)</b>	Script imports the created module and demonstrates each functionality	20 points
<b>Code Quality (20 points)</b>	Well-organized, commented, and error-free code	20 points

**Homework 3: Data Processing with Pandas and Visualization**

Description: Students are provided a dataset (e.g., CSV file). They will perform exploratory data analysis using pandas and visualize the results using matplotlib.

**Rubric:**

<b>Data Cleaning (20 points)</b>	Handle missing values, outliers, and any inconsistencies in the dataset	20 points
<b>Data Exploration (40 points)</b>	Use pandas operations to obtain insights, like mean, median, unique values, etc.	20 points
<b>Visualization (20 points)</b>	At least three different types of plots/graphs showcasing insights	20 points
<b>Analysis Summary (20 points)</b>	Written summary of findings from the dataset	20 points
<b>Code and Documentation Quality (20 points)</b>	Readable code with comments and proper documentation	20 points