DO NOT TYPE IN THIS BOX
Bulletin # : _____
Academic Year : _____

FLORIDA INTERNATIONAL UNIVERSITY
UNIVERSITY CURRICULUM COMMITTEE
Proposal for a New Course

1. **School/College** College of Engineering and Computing
   Div./Dept. in Which Taught Computing and Information Sciences

2. | COP | 2 | | | 3 |
   |---|---|---|---|---|
   | **Alpha Prefix** | **1st Digit** | **Last 3 Digits** | **"C"-lec-lab "L"-Lab** | **Cr. Hrs.** |

   CIP Code ***(Leave this blank):*** _____

3. **Grading Method (select one):** ☒ Graded ☐ Pass/Fail

4a. **Course Title** Python Programming I

b. **Abbreviated course Title** ***(for computer class schedules, transcripts)*** Python Programming I
LIMITED TO 25 Characters *(including spaces)*

5. **Statewide Course Numbering Subject Matter Area** Computer Science

6. **Catalog Description/Major Topics** *(not to exceed 200 characters including spaces)*
College of Medicine and College of Law: Attach description not exceeding 1,000 characters including spaces.

Introduction to computer programming using Python including fundamental concepts and systematic design techniques. Students will write programs that computationally solve and reduce problems.

7. **Attach detailed syllabus course outline and course justification on separate page(s).**
8. **Prerequisite(s):** MAC 1140 or MAC 1147 or MAC 2233 or MAC 2311 or Advisor's permission
9. **Corequisite(s):** _____
10. **Objective(s) of Course:**

Equip students with foundational Python programming skills and systematic design techniques. For a complete description and course outcomes, please refer to the attached syllabus.

11. **Does this course duplicate/overlap other courses at FIU?** ☒ No ☐ Yes
If yes, please explain: _____
12. **What other closely related department(s) have been consulted about this course?**
Department of Mathematics and Statistics
13. **Is this course used for the assessment of a program or a certificate (if yes, then send a notification to assessment@fiu.edu)?** ☑ **No** ☐ **Yes**

PROPOSAL REQUESTED BY:

Faculty Contact Gregory Murad Reis (Gregory Murad Reis) 09 / 04 / 20 23
(Type name) **(Signature)**
gmuradre@fiu.edu 305-348-7852
(Email address) **(Phone number)**

Chairperson (Dept./Div.) Jason Liu 10 / 10 / 20 23
(Type name) **(Signature)**

Chairperson (Curr. Comm.) _____ ____ / ____ / 20____
(Type name) **(Signature)**

College/School Dean _____ ____ / ____ / 20____
(Type name) **(Signature)**

Submit one original form. Attach one copy of the course justification and a draft of the course syllabus for this New Course Proposal. *The syllabus should include all components on the New Course Checklist.*

Justification for Python Programming I

In today's data-driven world, foundational programming skills have become an indispensable asset. "Python Programming I" is an essential cornerstone for the proposed **BS in Data Science** program at Florida International University (FIU). Python is usually the primary language for data analysis, and its inclusion as a core course ensures that students acquire not just theoretical knowledge, but also the practical skills needed to be successful in the rapidly evolving field of Data Science. This course offers a comprehensive introduction to Python programming, while emphasizing key concepts like systematic design techniques, effective utilization of data structures, and object-oriented principles. Each of the articulated course outcomes, from understanding core Python syntax to breaking down multifaceted problems into actionable solutions, aligns with the broader objectives of the BS in Data Science program. As FIU aims to produce top-tier data scientists capable of innovating and leading in the global data landscape, it is imperative that the curriculum is anchored with robust programming courses like "Python Programming I" to provide students with a strong foundation for their subsequent advanced studies and professional pursuits.

Knight Foundation School of Computing and Information Sciences

Course Title: Python Programming I **Date:** 09/18/2023

Course Number: COP 2XXX

Number of Credits: 3

Subject Area: Programming	**Subject Area Coordinator:** Janki Bhimani **email:** jbhimani@fiu.edu
Catalog Description: Introduction to computer programming using Python including fundamental concepts and systematic design techniques. Students will write programs that computationally solve and reduce problems.	
Textbooks: Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 3rd Edition, by Eric Matthes. Release: January 10, 2023. Publisher: No Starch Press. ISBN: 978-1718502703	
References (for further reading): Fluent Python, 2nd Edition, by Luciano Ramalho. Released April 2022. Publisher: O'Reilly Media, Inc. ISBN: 978-1492056355	
Prerequisites Courses: MAC 1140 or MAC 1147 or MAC 2233 or MAC 2311 or Advisor's permission	
Corequisite Courses:	

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

Prerequisites Topics:

1. Mathematical functions
2. Arithmetic and geometric sequences

Course Outcomes:

1. **Describe** the structure and characteristics of various data structures such as lists, tuples, dictionaries, and sets.
2. **Apply** the fundamentals tools of Python to create simple to moderately complex programs.
3. **Implement** data structures effectively to solve real-world problems.
4. **Recall** the core concepts and syntax of the Python programming language.
5. **Analyze** a given problem to identify appropriate data structures and algorithms for solutions.
6. **Develop** algorithmic solutions for common computational problems.
7. **Collaborate** writing, debugging, and refining Python code.
8. **Execute** programming projects from inception to completion, focusing on best practices and testing methodologies.
9. **Utilize** object-oriented programming principles, like encapsulation, in Python to design software.
10. **Break down** complex problems into manageable tasks or modules that can be addressed with Python functions or classes.

Knight Foundation School of Computing and Information Sciences
COP 2* Python Programming I**

Association between Student Outcomes and Course Outcomes

BS in Computing: Student Outcomes Graduates of the program will have an ability to:	Course Outcomes
1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.	1,2,3,4,5,6,7,8,9,10
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.	1,2,3,4,5,6,7,8,9,10
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.	

Program Specific Student Outcomes	
6) Apply theory, techniques, and tools throughout the data science lifecycle and employ the resulting knowledge to satisfy stakeholders' needs. [DS]	5,6,7,8,9,10

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 2* Python Programming I**

Outline

Topic	**Number of Lecture Hours** (Total: 37.5 hours = 15 weeks * 2 lectures/week * 1.25 hrs/lecture)	Outcome
1. Introduction to programming and Python 1.1. Introduction to programming, Python, and course overview 1.2. Python basics - variables, data types, and operators	3.75	2,4,7
2. Control structures 2.1. Conditionals (if, elif, else) 2.2. Loops (for, while)	3.75	2,4,7
3. Functions and modular programming 3.1. Functions - definition, arguments, return values 3.2. Scope and lifetime of variables	5	2,4,7
4. Data structures I - Lists and Tuples 4.1. Lists - operations, indexing, and slicing 4.2. Tuples and list comprehensions	5	1,3,4,5
5. String manipulation 5.1. String basics - operations, methods, and formatting 5.2. Reading and writing files (JSON)	5	2,4,7,8
6. Data structures II - Dictionaries and Sets 6.1. Dictionaries - operations, keys, and values 6.2. Sets and set operations	5	1,3,4,5,8
7. File I/O and exception handling 7.1. Error handling with try, except, and finally	3.75	2,4,6,7
8. Object-oriented programming I 8.1. Introduction to OOP, classes, and objects 8.2. Attributes, methods, and constructors	6.25	6,9,10

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 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%
	TOTAL	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: Python Basics and Control Structures

Description: Students will develop an interactive quiz application that poses multiple-choice questions to the user, records their answers, provides immediate feedback, and summarizes their performance at the end.

Description of Possible Rubric:

Criteria	Excellent (100)	Good (80)	Average (60)	Below Average (40)	Poor (20)	Weight
Code Structure						
- Organization and Modularity	Code is exceptionally organized with a clear modular structure	Code is well-organized with a good modular structure	Code has a basic organization with a moderate modular structure	Code has poor organization with limited modularity	Code is disorganized with no modular structure	15%
- Comments and Docstrings	Code contains comprehensive comments and detailed docstrings	Code contains good comments and docstrings	Code contains basic comments and some docstrings	Code contains few comments and minimal docstrings	Code contains no comments or docstrings	15%
Functionality						
- Question Prompt and Display	Questions are prompted and displayed perfectly with a	Questions are prompted and displayed well with a good user interface	Questions are prompted and displayed adequately with	Questions are prompted and displayed poorly with a	Questions are not prompted or displayed correctly	20%

	user-friendly interface		a basic user interface	limited user interface		
- Feedback on User's Answer	Provides perfect immediate feedback on the user's answer	Provides good immediate feedback on the user's answer	Provides basic immediate feedback on the user's answer	Provides poor immediate feedback on the user's answer	Does not provide feedback on the user's answer	15%
- Quiz Performance Summary	Summarizes quiz performance at the end perfectly with detailed feedback	Summarizes quiz performance at the end well with good feedback	Summarizes quiz performance at the end with basic feedback	Summarizes quiz performance at the end poorly with limited feedback	Does not summarize quiz performance at the end	10%
Control Structures						
- Use of Conditionals	Demonstrates excellent use of conditionals with efficient structures	Demonstrates good use of conditionals with effective structures	Demonstrates average use of conditionals with basic structures	Demonstrates below-average use of conditionals with limited structures	Demonstrates poor use of conditionals with incorrect or no structures	15%
- Understanding of Loops	Demonstrates excellent understanding of loops, using them efficiently to iterate through questions	Demonstrates good understanding of loops, using them effectively to iterate through questions	Demonstrates average understanding of loops, using them to iterate through questions	Demonstrates below-average understanding of loops, with limited use in iterating through questions	Demonstrates no understanding of loops, not using them to iterate through questions	10%

Homework 2: Data Structures in Action

Description: Students are to create a contact book program with search functionality. Besides adding, viewing, and deleting contacts, users should be able to search for contacts based on any field (name, phone, or email) and receive a list of matching results.

Description of Possible Rubric:

Criteria	Excellent (100)	Good (80)	Average (60)	Below Average (40)	Poor (20)	Weight
Code Structure						
- Modular and Maintainable Structure	Code is exceptionally modular and maintainable with a well-structured architecture	Code is well-organized with a good modular and maintainable structure	Code has a basic modular and maintainable structure	Code has a limited modular and maintainable structure	Code is not modular and lacks a maintainable structure	20%
- Comments and Docstrings	Code contains comprehensive comments and detailed docstrings	Code contains good comments and docstrings	Code contains basic comments and some docstrings	Code contains few comments and minimal docstrings	Code contains no comments or docstrings	15%
Functionality						
- Addition and Deletion of Contacts	Perfectly implements addition and deletion of	Implements addition and deletion of contacts well	Implements addition and deletion of contacts	Implements addition and deletion of contacts poorly	Does not implement addition and deletion of	15%

	contacts with user-friendly features	with good features	adequately with basic features	with limited features	contacts correctly	
- View Contacts	Displays contacts perfectly with a neat and user-friendly format	Displays contacts well with a good format	Displays contacts adequately with a basic format	Displays contacts poorly with a limited format	Does not display contacts correctly or at all	10%
- Search Contacts	Perfectly implements search functionality with relevant results for any field	Implements search functionality well with mostly relevant results for any field	Implements search functionality adequately with some relevant results for any field	Implements search functionality poorly with few relevant results for any field	Does not implement search functionality correctly or at all	10%
Control Structures						
- Handling Lists/Tuples for Storage	Demonstrates excellent proficiency in handling lists/tuples for storage with efficient structures	Demonstrates good proficiency in handling lists/tuples for storage with effective structures	Demonstrates average proficiency in handling lists/tuples for storage with basic structures	Demonstrates below-average proficiency in handling lists/tuples for storage with limited structures	Demonstrates poor proficiency in handling lists/tuples for storage with incorrect or no structures	15%
- Use of Dictionaries for Details and Search	Demonstrates excellent use of dictionaries for storing contact details and facilitating search with efficient structures	Demonstrates good use of dictionaries for storing contact details and facilitating search with effective structures	Demonstrates average use of dictionaries for storing contact details and facilitating search with basic structures	Demonstrates below-average use of dictionaries for storing contact details and facilitating search with limited structures	Demonstrates poor use of dictionaries for storing contact details and facilitating search with incorrect or no structures	15%

Homework 3: Object-Oriented Programming

Description: Students are required to model a library system using Python classes. This should include classes for Books, Users, and a Library. Users should be able to check out and return books.

Description of Possible Rubric:

Criteria	Excellent (100)	Good (80)	Average (60)	Below Average (40)	Poor (20)	Weight
Code Structure						
- Cleanliness and Organization	Code is impeccably clean, organized, and properly indented, showcasing a high level of professionalism	Code is clean, organized, and properly indented, showcasing good coding standards	Code is moderately clean, organized, and properly indented, showcasing average coding standards	Code is somewhat messy, less organized, and not properly indented, showcasing below-average coding standards	Code is messy, disorganized, and not properly indented, showcasing poor coding standards	15%

Knight Foundation School of Computing and Information Sciences
COP 2*** Python Programming I

- Use of Comments	Code contains comprehensive comments that perfectly explain complex code segments	Code contains good comments that explain most code segments well	Code contains basic comments that explain some code segments	Code contains few comments, with many segments left unexplained	Code contains no comments, leaving code segments unexplained	10%
Functionality						
- Check Out Books	Users can perfectly check out books with a user-friendly and error-free functionality	Users can check out books well with minor issues	Users can check out books, but the functionality has some issues	Users can barely check out books, with many issues in the functionality	Users cannot check out books, the functionality is broken or missing	15%
- Return Books	Users can perfectly return books with a user-friendly and error-free functionality	Users can return books well with minor issues	Users can return books, but the functionality has some issues	Users can barely return books, with many issues in the functionality	Users cannot return books, the functionality is broken or missing	15%
Object-Oriented Concepts						
- Use of Classes and Objects	Demonstrates excellent use of classes and objects, showcasing a deep understanding of OOP concepts	Demonstrates good use of classes and objects, showcasing a substantial understanding of OOP concepts	Demonstrates average use of classes and objects, showcasing a moderate understanding of OOP concepts	Demonstrates below-average use of classes and objects, showcasing limited understanding of OOP concepts	Demonstrates poor use of classes and objects, showcasing no understanding of OOP concepts	15%
- Methods within Classes	Demonstrates excellent use of methods within classes, showcasing a deep understanding of OOP concepts	Demonstrates good use of methods within classes, showcasing a substantial understanding of OOP concepts	Demonstrates average use of methods within classes, showcasing a moderate understanding of OOP concepts	Demonstrates below-average use of methods within classes, showcasing limited understanding of OOP concepts	Demonstrates poor use of methods within classes, showcasing no understanding of OOP concepts	15%
- Understanding of Attributes and Inheritance	Demonstrates excellent understanding of attributes and inheritance, applying them appropriately and effectively	Demonstrates good understanding of attributes and inheritance, applying them appropriately	Demonstrates average understanding of attributes and inheritance, with basic application	Demonstrates below-average understanding of attributes and inheritance, with limited application	Demonstrates poor understanding of attributes and inheritance, with incorrect or no application	15%