STATE UNIVERSITY OF NEW YORK
COLLEGE OF TECHNOLOGY
CANTON, NEW YORK

MASTER SYLLABUS

CYBR 181, Programming Fundamentals

Created by: Mehdi Ghayoumi
Updated by:

SCHOOL OF SCIENCE, HEALTH, & CRIMINAL JUSTICE
CENTER FOR CRIMINAL JUSTICE, INTELLIGENCE, & CYBERSECURITY
Fall 2024

A. **TITLE**: CYBR 181, Programming Fundamentals

B. **COURSE NUMBER:** CYBR 181

C. **CREDIT HOURS: 3**

D. **WRITING INTENSIVE COURSE**: n/a

E. **GER CATEGORY:** n/a

F. **SEMESTER(S) OFFERED:** Fall and Spring

G. **COURSE DESCRIPTION:** **Programming Fundamentals** is a foundational course designed to introduce students to the essentials of programming with a specific focus on Python, one of the most popular and versatile programming languages in contemporary computing and data science. This course is ideal for beginners with little to no prior experience in programming.
Course Objectives:

- **Introduction to Python**: Students will learn the basics of Python, including its syntax and key programming constructs.
- **Fundamental Programming Concepts**: The course covers essential programming principles such as variables, data types, control structures (loops and conditionals), and basic data structures like lists and dictionaries.
- **Practical Application**: Emphasis on hands-on learning where students write and execute Python scripts to solve real-world problems.
- **Functions and Modular Programming**: Introduction to writing reusable functions, understanding scope and parameter passing.
- **File Handling**: Basics of reading from and writing to files using Python, an essential skill for data processing.
- **Error Handling and Debugging**: Techniques to identify and fix errors in Python code, enhancing problem-solving skills.

H. **PRE-REQUISITES/CO-REQUISITES:**

a. Pre-requisite(s): None.

I. STUDENT LEARNING OUTCOMES:

Course Student Learning Outcome [SLO]	*PSLO ()*	*GER*	*ISLO*
a. **Recall** and **summarize** Python syntax and programming concepts.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
b. **Construct** basic data structures and **develop** simple algorithms in Python.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
c. **Apply** Python scripts to **analyze** and **interpret** data sets.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
d. **Evaluate** and **refine** Python programs for errors and performance.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
e. **Execute** file handling operations and **manage** I/O in Python.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
f. **Differentiate** between and **utilize** various control structures in Python.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
g. **Design** and **assemble** functions and modules for reuse in Python.			5. Industry, Professional, Discipline-Specific Knowledge and Skills
h. **Collaborate** using version control systems to **manage** changes in code.			5. Industry, Professional, Discipline-Specific Knowledge and Skills

KEY	**Institutional Student Learning Outcomes [ISLO 1 – 5]**
ISLO #	**ISLO & Subsets**
1	**Communication Skills** Oral [O], Written [W]
2	**Critical Thinking** *Critical Analysis [CA] , Inquiry & Analysis [IA] , Problem Solving [PS]*
3	**Foundational Skills** *Information Management [IM], Quantitative Lit,/Reasoning [QTR]*
4	**Social Responsibility** *Ethical Reasoning [ER], Global Learning [GL],* *Intercultural Knowledge [IK], Teamwork [T]*
5	**Industry, Professional, Discipline Specific Knowledge and Skills**

J. APPLIED LEARNING COMPONENT: **Yes______ No_X_____**

If Yes, select one or more of the following categories:

Classroom/Lab___
Internship___
Clinical Practicum___
Practicum___
Service Learning___

Civic Engagement___
Creative Works/Senior Project___
Research___
Entrepreneurship___
(program, class, project)

Community Service___

K. Suggested TEXTS:

1. **"Think Python: How to Think Like a Computer Scientist" by Allen B. Downey**
   - This book approaches Python from a computer science perspective, teaching students how to think critically and solve problems like a programmer.
2. **"Python Crash Course: A Hands-On, Project-Based Introduction to Programming" by Eric Matthes**
   - This book is ideal for beginners and provides a practical introduction to programming fundamentals using Python, covering concepts like data structures, loops, classes, and files.
3. **"Learning Python" by Mark Lutz**
   - A comprehensive guide to Python programming, this book covers Python in-depth, making it suitable for students as they progress through the course.
4. **"Python for Data Analysis" by Wes McKinney**
   - Although focused on data analysis, this book is excellent for understanding how Python can be applied in real-world scenarios, particularly in data science.

L. REFERENCES: n/a

M. EQUIPMENT: n/a

N. GRADING METHOD: A-F

O. SUGGESTED MEASUREMENT CRITERIA/METHODS:

- Participation Assignments
- Challenge Assignments
- Quizzes
- Exams

P. DETAILED COURSE OUTLINE:

Week 1: Introduction to Programming and Python
- Overview of programming concepts.
- Introduction to Python: Installation, setting up the environment.

Week 2: Python Basics
- Understanding Python syntax.
- Variables, data types, and basic operators.

Week 3: Control Structures
- Conditional statements (if, else, elif).
- Introduction to loops (for, while).

Week 4: Functions in Python
- Defining and calling functions.
- Understanding function arguments and return values.

Week 5: Data Structures (Part 1)
- Lists and list operations.
- Tuples and their usage.

Week 6: Data Structures (Part 2)
- Dictionaries and sets.
- Choosing the right data structure for a problem.

Week 7: String Manipulation
- String operations and functions.

- Introduction to string formatting.

Week 8: File Input/Output

- Reading from and writing to files.
- Understanding file paths and working with different file formats.

Week 9: Error Handling and Exceptions

- Basic error handling using try-except blocks.
- Raising exceptions.

Week 10: Introduction to Modules and Packages

- Using built-in modules.
- Installing and using external packages.

Week 11: Working with APIs

- Basics of web APIs.
- Fetching data from APIs using Python.

Week 12: Introduction to Data Analysis Libraries

- Overview of pandas and NumPy.
- Basic data manipulation and analysis.

Week 13: Basic Data Visualization

- Introduction to matplotlib and seaborn.
- Creating simple plots and charts.

Week 14: Practical Applications of Python in Data Science

- Overview of real-world applications.
- Mini-project: Applying Python skills to solve a data science problem.

Week 15: Course Review and Project Presentations

- Review of key concepts.
- Presentation of mini-projects.

Q. LABORATORY OUTLINE:

n/a