



Python Certified Associate Programmer Certification (PCAP)
CN906

Hours: In Class 69 Clinical 0 Total 69

Description

#1 Programming Language in the world as ranked by IEEE. In this course, you will learn to design, write, debug, and run programs encoded in the Python language. No prior programming knowledge is required. Python is a general-purpose programming language used to build just about anything. Python is key for back-end web development, data analysis, artificial intelligence and scientific computing, all of which are key for pursuing IT careers. The course aligns to the PCAP - Python Certified Associate Programmer certification, validating your expertise to employers and expanding your IT and IoT career opportunities. When you complete the course, you are eligible for a 51% discount against the exam fee that reduces the exam to \$144.55. The exam includes a free retake.

Prerequisites

Students should be familiar with personal computers and the use of a mouse and keyboard.

Books

- ISBN:

Learning Objectives

At the end of the course students will be able to:

1. Explain the fundamentals of computer programming, i.e. how the computer works, how the program is executed, how the programming language is defined and constructed, what the difference is between compilation and interpretation, what Python is, how it is positioned among other programming languages, and what distinguishes the different versions of Python.
2. Demonstrate the basic methods of formatting and outputting data offered by Python, together with the primary kinds of data and numerical operators, their mutual relations and bindings; the concept of variables and variable naming conventions; the assignment operator, the rules governing the building of expressions; the inputting and converting of data.
3. Distinguish Boolean values to compare difference values and control the execution paths using the if and if-else instructions; the utilization of loops (while and for) and how to control their behavior using the break and continue instructions; the difference between logical and bitwise operations; the concept of lists and list processing, including the iteration provided by the for loop, and slicing; the idea of multi-dimensional arrays.
4. Explain the defining and using of functions and their rationale, purpose, conventions, and traps; the concept of passing arguments in different ways and setting their default values, along with the mechanisms of returning the function's results; name scope issues; new data aggregates: tuples and dictionaries, and their role in data processing.
5. Explain Python modules: their rationale, function, how to import them in different ways, and present the content of some standard modules provided by Python; the way in which modules are coupled together to make packages; the concept of an exception and Python's implementation of exceptions, including the try-except instruction, with its applications, and the raise instruction; strings and their specific methods, together with their similarities and differences compared to lists.
6. Explain the fundamentals of OOP (Object Oriented Programming) and the way they are adopted in Python, showing the difference between OOP and the classical, procedural approach; the standard objective features: inheritance, abstraction, encapsulation, and polymorphism, along with Python-specific issues like instance vs. class variables, and Python's implementation of inheritance; objective nature of exceptions; Python's generators (the yield instruction) and closures (the lambda keyword); the means Python developers can use to process (create, read, and write) files.

Teaching Philosophy

We believe that instructors, staff, and administrators have a shared responsibility to provide: 1) innovative course design and instruction; 2) a safe, learner-centered environment; and 3) an authentic learning experience.

Teaching Methods

Methods include lecture, class discussion and demonstrations.

Evaluation Methods

Student success is based on participation in class activities and the completion of exercises. A certificate of completion requires successful completion of all assigned work within the established time frame. Types of graded assignments will be projects, review questions, activities, assignments and tests

Grading Policy:

A = 90 - 100%

B = 80 - 89%

C = 70 - 79%

D = 60 - 69%

F = Below 60%

A course grade of D does not qualify the course as a prerequisite to other courses.

Grading Policy

A certificate of completion requires a minimum of 80% attendance and completion of all assigned work.

Student Responsibilities

To ensure a quality and safe learning environment, students are required to follow the Post-Secondary Student Behavior policy #560. This policy can be found at www.mntc.edu/board-policies. Printed copies are available upon request.

Students are expected to attend class and participate in class discussions and activities, complete out of class assignments and exams in class.

Students must be on time and meet the attendance policy set for this class which is 80% attendance.