

## 55284A: INTRODUCTION TO PYTHON

### Course Outline

#### Module 1: Python Basics

Python, which first appeared in 1991, is one of the most popular programming languages in use. Python is a high-level programming language, meaning that it uses a syntax that is relatively human readable, which gets translated by a Python Interpreter into a language your computer can understand. Examples of other popular high-level programming languages are C#, Objective-C, Java, PHP, and JavaScript. Interestingly, all of these other languages, unlike Python, share a C-like syntax. If you use one or more of those languages, you may find Python's syntax a little strange. But give it a little time. You'll find it's quite programmer friendly.

#### Lessons

- Running Python
- Hello, World!
- 
- Literals
- Python Comments
- Variables
- WRiting a Python Module
- print() Function
- Collecting User Input
- Getting Help

#### Lab : Exercises in this Lesson

- Hello, World
- Exploring Types
- A Simple Python Script

After completing this module, students will be able to:

- Understand how Python works.
- Understand Python's place in the world of programming languages.
- Understand the difference between Python 3.x and Python 2.x.
- Understand variables and Python's data types.
- Create simple modules.
- Get help on Python

#### Module 2: Functions and Modules

This module explains how to work with functions and modules.

#### Lessons

- Defining Functions
- Variable Scope
- Global Variables
- Function Parameters
- Returning Values
- Importing Modules

### **Lab : Exercises in this Lesson**

- A Function with Parameters
- Parameters with Default Values

After completing this module, students will be able to:

- Define and call functions.
- Define parameters in the pass arguments to functions.
- Set default values for parameters.
- Understand variable scope.
- Return values from functions.
- Create and import modules.

### **Module 3: Math**

Python includes some built-in math functions and some additional built-in libraries that provide extended math (and related) functionality. In this lesson, we'll cover the built-in functions and the math and random libraries.

#### **Lessons**

- Arithmetic Operators
- Assignment Operators
- Built-in Math Functions
- The math Module
- The random Module

### **Lab : Exercises in this Lesson**

- Floor and Modulus
- How Many Pizzas Do We Need

After completing this module, students will be able to:

- Do basic math in Python.
- Use the math module for additional math operations.
- Use the random module to generate random numbers.

### **Module 4: Python Strings**

Python includes some built-in math functions and some additional built-in libraries that provide extended math (and related) functionality. In this lesson, we'll cover the built-in functions and the math and random libraries.

#### **Lessons**

- Quotation Marks and Special Characters
- String Indexing
- Slicing Strings
- Concatenation and Repetition
- Common String Methods
- String Formatting
- Formatted String Literals (f-strings)
- Built-in String Functions

### **Lab : Exercises in this Lesson**

- Indexing Strings
- Slicing Strings
- Repetition
- Playing with Formatting
- Getting Acquainted with f-strings
- Outputting Tab-delimited Text

After completing this module, students will be able to:

- Work with strings.
- Escape special characters
- Work with multi-line strings.
- Index and slice strings.
- Use common string operators and methods.
- Format strings.
- Use built-in string functions

### **Module 5: Iterables: Sequences, Dictionaries, and Sets**

Iterables are objects that can return their members one at a time. The iterables we will cover in this lesson are lists, tuples, ranges, dictionaries, and sets.

#### **Lessons**

- Definitions
- Sequences
- Unpacking Sequences
- Dictionaries
- The len() Function
- Sets
- \*args and \*\*kwargs

### **Lab : Exercises in this Lesson**

- Remove and Return Random Element
- Simple Rock, Paper, Scissors Game
- Slicing Sequences
- Creating a Dictionary from User Input

After completing this module, students will be able to:

- Understand the different types of iterables available in Python
- Create, modify, and work with lists.
- Create and work with tuples.
- Create ranges.
- Create, modify, and work with dictionaries
- Create sets and to use sets to remove duplicates in lists
- Understand the \*args and \*\*kwargs parameters.

### **Module 6: Flow Control**

Generally, a program flows line by line in sequential order. We have seen already that we can change this flow by calling functions. The flow can also be changed using conditional statements and loops.

## **Lessons**

- Conditional Statements
- Loops in Python
- break and continue
- The enumerate() Function
- Generators
- List Comprehensions

## **Lab : Exercises in this Lesson**

- All True and Any True
- Word Guessing Game
- Find the Needle
- Rolling Dice

After completing this module, students will be able to:

- Work with if conditions in Python.
- Work with loops in Python.
- Create generator functions
- Work with list comprehensions

## **Module 7: Virtual Environments**

A virtual environment is a container for a Python development platform on your computer. Packages can be installed in the virtual environment that are separate from your standard, or non-virtual environment.

## **Lessons**

- Virtual Environment

## **Lab : Exercises in this Lesson**

- Working with a Virtual Environment

After completing this module, students will be able to:

- Understand what a virtual environment is.
- Create a virtual environment.
- Activate and deactivate a virtual environment.
- Delete a virtual environment.

## **Module 8: Regular Expressions**

Regular expressions are used to do pattern matching in many programming languages, including, Java, PHP, JavaScript, C, C++, and Perl. We will provide a brief introduction to regular expressions and then we'll show you how to work with them in Python.

## **Lessons**

- Regular Expression Syntax
- Python's Handling of Regular Expressions

After completing this module, students will be able to:

- Create regular expressions for pattern matching.
- Use regular expressions within Python.

## **Module 9: Unicode and Encoding**

This module explains how to work with Unicode and encoding.

### **Lessons**

- Bits and Bytes
- Hexadecimal Numbers
- Encoding

### **Lab : Exercises in this Lesson**

- Converting Numbers between Number Systems
- Finding Confusables

After completing this module, students will be able to:

- Understand binary and hexadecimal numbers.
- Convert numbers between number systems.

## **Module 10: File Processing**

Python allows you to access and modify files and directories on the operating system. Among other things, you can: 1. Open new or existing files and store them in file object variables. 2. Read file contents, all at once or line by line. 3. Append to file contents. 4. Overwrite file contents. 5. List directory contents. 6. Rename files and directories.

### **Lessons**

- Opening Files
- The os and os.path Modules

### **Lab : Exercises in this Lesson**

- Finding Text in a File
  - Writing to Files
  - Listing Creator
- After completing this module, students will be able to:
- Read files on the operating system.
  - Create and write to files on the operating system.
  - Access and work with directories on the operating system.
  - Work with the os and os.path modules.

## **Handling**

This module explains how to handle exceptions in Python.

### **Lessons**

- Wildcard except Clauses
- Getting Information on Exceptions
- The else Clause
- The finally Clause
- Using Exceptions for Flow Control
- Raising your Own Exceptions
- Exception Hierarchy

### **Lab : Exercises in this Lesson**

- Raising Exceptions
- Running Sum

After completing this module, student will be able to:

- Handle exceptions in Python.

## **Module 12: Python Dates and Times**

This module explains how to work with Python dates and times.

### **Lessons**

- Understanding Time
- The time Module
- The datetime Module

### **Lab : Exercises in this Lesson**

- Report on Amtrack Departure Times

After completing this module, students will be able to:

- Work with the time module.
- Work with the datetime module.

## **Module 13: Running Python Scripts from the Command Line**

This module explains how to run Python scripts from the command line.

### **Lessons**

- sys.arg
- sys.path

### **Lab : Exercises in this Lesson**

- Migrating Settings by using Windows Easy Transfer
- Configuring a Reference Image of Windows 7
- Configuring a Reference Image

After completing this module, students will be able to:

- Run Python scripts from the command line.

## **Additional Reading**

None