

Python Programming

Course ISI-1481B 5 Days Instructor-led, Hands-on

Course Description

This five-day instructor led course provides the fastest and easiest, yet most professional way to learn Python, whether you are a beginning programmer or have years of experience.

Python is one of today's most popular languages. Its simple syntax makes it relatively easy to learn, and its diverse uses...from web and game programming to data analysis and data mining to scientific computing, artificial intelligence, and more...have created a huge demand for programmers.

If Python is your first programming language, it will help you master Python, of course. But it will also help you master the skills and concepts you need to program in any modern language.

If you're an experienced programmer who wants to add Python to your resume, it will help you learn Python faster and better than you've ever learned a language before.

Course Objectives

Upon successful completion of this course, students understand

- Gain a strong set of the base skills that you'll use in any Python program
- Learn about working with numbers and strings
- Work with dates and times, dictionaries, recursion and algorithms
- Learn how to design and develop object-oriented programs
- Apply the skills learned to real-world programs
- Use Python for database programming
- Build on that base by learning the other essential skills that every Python programmer should have
- Develop object-oriented database and GUI programming
- Apply all the skills that you've learned as you build database and GUI programs for the real world.
- Use Python to develop programs with graphical user interfaces (GUIs)

Prerequisites

Students should have some experience with at least one programming language. Typically, students in this course will have already programmed in either C, C++, Java, Perl, Ruby, VB, or anything equivalent to these languages.

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

Course Outline

Module 1: An Introduction to Python Programming

- An introduction to Python programming
- Why Python works so well as your first programming language
- Three types of Python applications
- The source code for a console application
- How Python compiles and runs source code
- How disk storage and main memory work together
- How to use IDLE to develop programs
- How to use the interactive shell
- How to work with source files
- How to compile and run a program
- How to fix syntax and runtime error

Module 2: How to write your first programs

- Basic coding skills
- How to code statements
- How to code comments
- How to use functions
- How to work with data types and variables
- How to assign values to variables
- How to name variables
- How to work with numeric data
- How to code arithmetic expressions
- How to use arithmetic expressions in assignment statements
- How to use the interactive shell for testing numeric operations

Module 3: How to Code Control Statements

- How to code Boolean expressions
- How to use the relational operators
- How to use the logical operators
- How to compare strings
- How to code the selection structure
- How to code if statements
- More examples of if statements
- How to code nested if statements
- How to use pseudocode to plan if statements
- How to use the iteration structure
- How to code while statements
- How to code for statements
- How to code break and continue statements

- More examples of loops
- How to use assignment expressions
- How to use pseudocode to plan a program

Module 4: How to define and use functions and modules

- How to define and call a function
- How to define and call a main() function
- The Future Value program with functions
- More skills for defining and using functions
- How to use default values for arguments
- How to use named arguments
- When and how to use local and global variables
- How to create and use modules
- How to create a module
- How to document a module
- How to import a module
- The Convert Temperatures program
- How to use standard modules
- How to use the random module
- How to plan the functions of a program
- How to use a hierarchy chart
- The hierarchy chart for the Pig Dice game

Module 5: How to test and debug a program

- An introduction to testing and debugging
- The three types of errors that can occur
- Common Python errors
- Four techniques for testing and debugging
- How to plan the test runs
- A simple way to trace code execution
- How to use top-down coding and testing to simplify debugging
- How to use the IDLE shell to test functions
- How to use the IDLE debugger
- How to set and remove breakpoints
- How to step through the code
- How to view the stack

Module 6: How to work with lists and tuples

- Basic skills for working with lists
- How to create a list
- How to get and set items
- How to add and remove items

- How to process the items in a list
- More skills for processing the items in a list
- How lists are passed to functions
- How to work with a list of lists
- How to create a list of lists
- How to process the items in a list of lists
- More skills for working with lists
- How to count, reverse, and sort the items in a list
- How to get minimum, maximum, sum, and random values from a list
- How to copy, slice, and concatenate lists
- How to map, filter, and reduce the items in a list
- How to work with list comprehensions
- How to work with tuples
- How to create a tuple
- How to get items from a tuple

Module 7: How to work with file/O

- An introduction to file I/O
- How file I/O works
- How to open and close a file
- How to use text files
- How to write a text file
- How to read a text file
- How to work with a list in a text file
- How to use CSV files
- How to write a CSV file
- How to read a CSV file
- How to modify the CSV format
- How to use binary files
- How to work with a binary file
- The Movie List 3.0 program
- How to use binary files
- How to work with a binary file
- The Movie List 3.0 program

Module 8: How to handle exceptions

- How to handle a single exception
- How exceptions work
- How to use a try statement to handle one type of exception
- How to handle multiple exceptions
- How to use a try statement to handle multiple exceptions
- How to get the information from an exception object
- Two more skills

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- How to use a finally clause
- How to raise an exception

Module 9: Other Concepts and Skills

- Basic skills for working with numbers
- How floating-point numbers work
- How to use the math module
- How to format numbers
- How to use format specifications with f-strings
- How to use the locale module
- How to fix rounding errors
- How to work with decimal numbers
- How to use the decimal module
- The Invoice program with decimal numbers

Module 10: How to work with strings

- Basic skills for working with strings
- Unicode, indexes, slicing, duplicating, and multiline strings
- How to search a string
- How to loop through the characters in a string
- How to use basic string methods
- How to find, remove, and replace parts of a string
- How to split and join strings
- How to split a string into a list of strings
- How to join strings

Module 11: How to work with dates and times

- How to get started with dates and times
- How to create date, time, and datetime objects
- How to create datetime objects by parsing strings
- How to format dates and times
- How to work with spans of time
- More skills for working with dates and times
- How to get date and time parts
- How to compare date/time objects

Module 12: How to work with dictionaries

- How to get started with dictionaries
- How to create a dictionary
- How to get, set, and add items
- How to delete items
- How to loop through keys and values

- How to convert between dictionaries and lists
- More skills for working with dictionaries
- How to use the merge and update operators
- How to use dictionaries with complex objects as values

Module 13: How to work with recursion and algorithms

- An introduction to recursion
- How recursion works in Python
- How to use recursion to add a range of numbers
- Some common recursive algorithms
- How to compute the factorial of a number
- How to compute a Fibonacci series

Module 14: How to define and use your own classes

- An introduction to classes and objects
- Two UML diagrams for the Product class
- Code that defines a Product class
- Code that uses a Product class
- How to create and use objects
- How to define a class
- How to code a constructor and attributes
- How to code methods
- How work with object composition
- How object composition works
- The Die and Dice classes
- How to work with encapsulation
- How object encapsulation works
- How to hide attributes
- How to access hidden attributes with methods
- How to access hidden attributes with properties
- The Die and Dice classes with encapsulation
- The Product class with some encapsulation

Module 15: How to work with inheritance

- How to work with inheritance
- How inheritance works
- How to define a subclass
- How polymorphism works
- How to check an object's type
- How to override object methods
- How to define a string representation for an object
- How to define an iterator for an object

- The Die and Dice classes
- Two more skills for the road
- How to work with custom exceptions
- When to use inheritance

Module 16: How to design an object-oriented program

- Techniques for object-oriented design
- Five steps for designing an object-oriented program
- How to identify the data attributes
- How to subdivide the data attributes
- How to identify the classes
- How to identify the methods and properties
- How the three-tier architecture works
- The Shopping Cart program
- The business tier
- The database tier
- The presentation tier

Module 17: How to work with a database

- An introduction to relational databases
- How a database table is organized
- How the tables in a database are related
- How the columns in a table are defined
- How to use the SQL statements for data manipulation
- How to select data from a single table
- How to select data from multiple tables
- How to insert, update, and delete rows
- How to use DB Browser for SQLite to work with a database
- How to use DB Browser to view and edit a table in a SQLite database
- How to use DB Browser to run SQL statements
- How to use Python to work with a database
- How to connect to a SQLite database
- How to execute SELECT statements
- How to get the rows in a result set
- How to execute INSERT, UPDATE, and DELETE statements
- How to test the database code
- How to handle database exceptions

Module 18: How to build a GUI program

- How to create a GUI that handles an event
- How to display a root window
- How to work with frames and buttons



- How to handle a button click event
- More skills for working with components
- How to work with labels and text entry fields
- How to lay out components in a grid
- How to code a class that defines a frame
- How to display a message box