Sophia Learning

CS1100: Introduction to Programming With Python (3 semester credits)

COURSE DESCRIPTION

In this course, you will learn the basics of computer programming from data types, to creating classes, to algorithms and testing. You will learn these concepts while diving deep into the syntax of Python as your core programming language. The course culminates in the chance to design and build a project that answers a need or goal that you determine.

COURSE EFFECTIVE DATES: June 2022 - Present

PREREQUISITES: No prerequisites

LENGTH OF COURSE: This is a self-paced course. Students may use as much or as little time as needed to complete the course.

ACE CREDIT® RECOMMENDATION: In the lower-division baccalaureate/associate degree category, 3 semester hours in introduction to programming.

GRADING: This is a pass/fail course. Students must complete 11 Challenges (formative assessments), 3 Milestones (summative assessments), and 1 Touchstone (project-based or written assessments) with an overall score of 70% or better.

LEARNING OUTCOMES

Upon completion of the course, the student will be able to:

1. Describe the steps a programmer goes through to tackle a new problem.
2. Recall the most common data types and how to create and use them in a program.
3. Insert conditional statements and functions into a program.
4. Create and manipulate lists and other common data collection types.
5. Use loops to repeat steps either a fixed number of times or dynamically based on conditions.
6. Develop and test complex functions including special Python functions.
7. Write the code for a basic class.
8. Remember the situations where inheritance and scope will affect a program's operations.
9. Use modules and files to add functionality and data to a program.

10. Plan an algorithm for an original program.
11. Code and test an original program based on a prepared algorithm.

OUTLINE OF MAJOR CONTENT AREAS

Unit 1: Program Basics

- Challenge 1: Learning to Code

1. Programming Mindset
2. Thinking Through Examples
3. Forming an Algorithm
4. Guide to Using Replit
5. Testing

- Challenge 2: Data Types

1. Data Types Introduction
2. Using Integers and Floats
3. Operators and Operands
4. Using Strings
5. String Operations
6. Debugging Operations
7. Coding Your First Program

- Challenge 3: Functions and Conditions

1. Functions and Methods Introduction
2. Conditional Statements
3. Boolean Operators
4. Multiple Conditions
5. Exceptions
6. Debugging Conditional Statements
7. Drink Order Program

Unit 2: Lists and Loops

- Challenge 1: Lists

1. Introduction to Lists
2. Manipulate Lists
3. Sets, Tuples, and Dictionaries
4. Multiple Dimensions
5. Debugging Lists
6. Tic-Tac-Toe Program

- Challenge 2: Loops

1. Introduction to Loops
2. Loops Using while
3. Loops Using for
4. Nested Loops
5. Debugging Loops
6. Revisiting the Tic-Tac-Toe Program

- Challenge 3: Complex Functions

1. Function Arguments
2. The Return Statement
3. Nested Functions
4. Debugging Functions
5. Finishing the Tic-Tac-Toe Program

Unit 3: Classes

- Challenge 1: Class Basics

1. Introduction to Classes
2. The init Method and del Function
3. Object Attributes
4. Object Methods
5. The Employee Class Program

- Challenge 2: Inheritance and Scope

1. Introduction to Inheritance
2. Subclasses
3. Scope
4. Class Troubleshooting
5. Revisiting the Employee Class Program

- Challenge 3: Modules and Files

1. Introduction to Modules
2. Using Modules
3. Creating Modules
4. Introduction to File I/O
5. Reading and Writing to a File
6. Common Issues with Files
7. Finishing the Employee Class Program

Unit 4: Project

- Challenge 1: Planning the Algorithm

1. Python Touchstone Overview
2. Identifying a Problem to Solve

3. Working An Example
4. Identifying the Patterns
5. Forming the Algorithm

- Challenge 2: Coding the Algorithm

1. Translating to Code
2. Writing the Program
3. Testing As You Go
4. Commenting Your Code
5. Course Wrap-Up

