

Course Code: CIS 122

Course Title: Introduction to Computer Programming

Department: Business/Computer Science and Technologies

Effective Date: Summer 2026

PCS Code: 1.1 - Baccalaureate/Transfer

CIP Code: 11.0201

Repeatability: 0

Credit Hours

Catalog Notation: 3-2-4

Credit Hour Distribution:

Lecture: 3

Lab: 2

Clinical: 0

Total: 4

General Course Information

Catalog Description

Introduction to programming logic and concepts with emphasis on syntax, structure and real-world applications. Use of an Integrated Development Environment (IDE). For students in science, mathematics, or technical programs.

General Course Objectives

- To learn and apply fundamental programming principles using the Python programming language.
- To understand and use basic computer science topics including data types, expressions, decision and loop structures, functions, sequences, and object-oriented programming.

Minimum Placement Levels

English

None

Reading

None

Math

Placement out of MAT 072

Prerequisites

None

Methods of Evaluation

8-10 lab programming assignments, 6-8 quizzes, and final exam.

Instructional Materials and Additional Supplies

Python Programming: An Introduction to Computer Science, 3rd edition, by John Zelle. 978-1-59028-275-5

Course Content

General Learning Outcomes (GLOs)

- Reasoning and Inquiry: Students will demonstrate the ability to solve problems using deductive reasoning and logic, quantitative reasoning, or the scientific method.
- Technology: Students will demonstrate the ability to evaluate, select, and appropriately use current and emerging tools.

Course Segments and Student Learning Outcomes

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Course Introduction	1. Identify course requirements, goals, expectations, and course information.	1	0	0
Computers and Programs	1. Describe and identify components of computer systems and relationships between hardware and software. 2. Describe the basic areas of study and techniques used in computer science. 3. Describe the form and function of computer programming languages with a focus on the Python programming language. 4. Demonstrate installation and operation of the Python integrated development environment (IDLE). 5. Apply basic Python language instructions by constructing a simple program in the development environment.	4	2	0
Writing Simple Programs	1. Identify and describe the major steps in an efficient software design process including analysis, specifications, design, testing, and debugging. 2. Identify and describe rules for forming valid Python identifiers, names, and expressions. 3. Identify and describe statements to output information, assign values to variables, and obtain input information from a user. 4. Apply basic Python language instructions using variables and expressions to solve a problem using input from a user within a program.	8	5	0
Computing with Numbers	1. Describe the concept of data types and their use in a program. 2. Identify and describe the basic numeric data types in Python and the fundamental principles of how numbers are represented in a computer. 3. Describe the concept of a Python function, libraries of functions, and their inclusion in a program with an emphasis on the Python math library. 4. Describe how a variable can be used to accumulate values within a looping mechanism. 5. Plan, design, and develop a program requiring the steps of gathering user input, using variables, applying expressions with the math library, and printing results to solve a problem statement.	8	5	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Objects and Graphics	<ol style="list-style-type: none"> 1. Describe the concept of objects and how they can be used to simplify program design and development; and distinguish between a class and an object. 2. Describe and distinguish between primitive data types and Python class types. 3. Identify and describe different classes available in a Python library to draw two-dimensional graphics. 4. Describe the fundamental concepts of working with graphical class objects from the Python library. 5. Design and develop a basic program demonstrating object-oriented principles using class objects from a Python library to draw a simple graphical representation in a window. 6. Describe the object-oriented principles of using and working with class methods, accessors, mutators, and aliasing. 7. Plan, design, and develop a program applying class methods, accessors, mutators, and aliasing to graphically represent a given concept from a problem statement within a graphics window. 	8	5	0
Sequences: Strings, Lists, Files	<ol style="list-style-type: none"> 1. Describe and identify the string data type and how strings are represented in the computer. 2. Describe the various operations that can be performed on strings using built-in Python functions and various methods of the string class. 3. Plan, design, and develop a program requiring strings, string methods, and user input to solve a problem statement. 4. Describe the basic concept of a sequence or list of elements as they apply to strings and Python lists. 5. Describe basic file-processing concepts and techniques for reading and writing text files in Python. 	8	5	0
Defining Functions	<ol style="list-style-type: none"> 1. Describe the use and purpose of functions in programming. 2. Describe the process of defining and creating a user-defined function in Python. 3. Describe the process and details of function calls and passing parameters to functions. 4. Plan, design, and develop a program requiring the definition of user-defined functions, function parameters, and function calls to solve a series of tasks within a programming description. 	2	2	0
Decision Structures	<ol style="list-style-type: none"> 1. Describe the basic concept of making decisions with boolean conditions in a computer program. 2. Describe and identify methods of constructing two-way decision statements in Python. 3. Describe and identify methods of constructing multi-way decision statements in Python. 4. Plan, design, and develop a program requiring the implementation of decision structures to solve a condition(s) within a programming description. 	3	3	0

Course Segment	Learning Outcomes	Lecture Hours	Lab Hours	Clinical Hours
Loop Structures and Booleans	<ol style="list-style-type: none"> 1. Describe the concepts of looping mechanisms including definite and indefinite loops. 2. Identify and describe programming patterns for interactive loops and sentinel loops and their implementations in Python. 3. Describe the purpose and function of nested loops. 4. Plan, design, and develop a program requiring the construction of looping mechanisms to implement an interactive looping program to solve tasks given in a problem statement. 	3	3	0

Total Contact Hours

Lecture Hours	Lab Hours	Clinical Hours
45	30	0