



## Course Change-Credit: CSIT 175 - Python Programming

Active

Fall Semester 2020

### Basic Course Information

Courses numbered 1 - 49 are remedial or college preparatory courses which do not apply toward an A. A. Degree and are not intended for transfer. Courses numbered 50-99 apply toward an AA Degree, but are not intended for transfer. Courses numbered 100 and higher apply toward an AA Degree and/or are intended for transfer to a four-year college or university.

**CB04 Course Credit Status** D - Credit - Degree Applicable

**Discipline** CSIT

**Title** Python Programming

**This course is part of the following cross listing**

### Units and Hours

**Min Unit Hour** 3.000

**Max Unit Hour** 3.000

**Grade Option** Grade/Pass/No Pass

#### Min Semester Hours Calculated

Hour Type	Weekly Hours		
Semester Hours x 16 Weeks	Semester Hours x 18 Weeks		
Lecture Category -	2.00	x 16 Weeks - 32.00	x 18 Weeks - 36.00
Lab Category -	3.00	x 16 Weeks - 48.00	x 18 Weeks - 54.00
Subtotal -	5.00	x 16 Weeks - 80.00	x 18 Weeks - 90.00
Out of Class Hour -	4.00	x 16 Weeks - 64.00	x 18 Weeks - 72.00
Totals -	9.00	x 16 Weeks - 144.00	x 18 Weeks - 162.00

#### Max Semester Hours Calculated

Hour Type	Weekly Hours		
Semester Hours x 16 Weeks	Semester Hours x 18 Weeks		
Max Lecture Category -	2.00	x 16 Weeks - 32.00	x 18 Weeks - 36.00
Max Lab Category -	3.00	x 16 Weeks - 48.00	x 18 Weeks - 54.00
Max Subtotal -	5.00	x 16 Weeks - 80.00	x 18 Weeks - 90.00
Max Out of Class Hour -	4.00	x 16 Weeks - 64.00	x 18 Weeks - 72.00
Max Totals -	9.00	x 16 Weeks - 144.00	x 18 Weeks - 162.00

**Basic Skills Requirements:** Appropriate Language and/or Computational Skills.

## Catalog Description

---

Provides the knowledge and skills necessary to use the Python programming language to develop software for Desktop and Web applications Topics of study include the fundamentals of the language, using lists, dictionaries and creating functions and classes

## Student Learning Outcomes

---

### Outcome

1. Industry Standards: Demonstrate Industry-accepted coding standards
2. Use and Test Software: Students will be able to use a requirements specification to design and develop Python programs that meet those requirements.

## Specific Course Objectives

---

### Objective

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

1. Understand the fundamentals of the Python programming language ;
2. Write syntactically and semantically correct Python programs;
3. Create Python Desktop and Web applications;
4. Analyze program requirements and design solutions using Python.

## Methods of Instruction

---

Methods of Instruction may include, but are not limited to, the following:

- Discussion
- Lab
- Lecture

## Content in Terms of Specific Body of Knowledge

---

- I. Variables and Data Types
  - A. Naming and Using Variables
  - B. Changing Case with Strings
  - C. Concatenation
  - D. Number Handling
- II. Working with Lists

- A. Changing, Adding, and Removing Elements
  - B. Organizing a List
  - C. Looping Through a List
  - D. Tuples
- III. Decision Statements
  - A. Conditional Tests
  - B. IF statements
  - C. Using IF statements with Lists
- IV. Dictionaries
  - A. Accessing Values in a Dictionary
  - B. Looping through Dictionary
  - C. Nesting
- V. User Input and Loops
  - A. Using the input() Function
  - B. While loops
  - C. Using a Loop with Lists and Dictionaries
- VI. Functions
  - A. Defining a Function
  - B. Passing Arguments
  - C. Return Values
  - D. Passing a List
  - E. Storing Functions in Modules
  - F. Styling Functions
- VII. Classes
  - A. Creating and Using a Class
  - B. Classes and Instances
  - C. Inheritance
  - D. Importing Classes
  - E. The Python Standard Library
- VIII. Files and Exceptions
  - A. Reading from a File
  - B. Writing to a File
  - C. Exceptions
  - D. Storing Data
- IX. Testing Code
  - A. Testing a Function
  - B. Testing a Class
- X. Web Programming with Python
  - A. Using Web Frameworks
  - B. Using REST API
  - C. Other Methods

## Textbooks/Resources

---

### Textbook

1. **Author** Urban, Michael and Murach, Joel  
**Title** Murach's Python Programming

**Edition** 1st**City** Fresno**Publisher** Mike Murach & Associates**Year** 2016

## Assignments

---

Students are expected to research projects and solutions as dictated by specific assignments. Students are expected to spend a minimum of three hours per unit per week in class and on outside assignments, prorated for short-term classes.

### **Required Reading**

Students will be required to read the textual material provided on the class web site and/or from the required book.

### **Suggested Reading**

### **Required Writing**

Students will explain what troubleshooting approaches they found most useful and share those findings with the other students in the class.

### **Critical Thinking**

All of the courses in computer programming require students to learn to think critically. Using computers effectively requires that students can express their instructions in a form that the computer program can understand and execute. First students must understand what they want to accomplish. Second, students must identify what logical steps are required to instruct the computer to do the work. Finally, students must know how to submit the instructions in the correct format to the computer for completion. Accomplishing this work requires students apply critical thinking skills.

### **Outside Assignments**

Outside assignments will include completion of lab work, assigned readings the text books, and homework problems.

## Methods of Assessment

---

### Evaluation Method

- Class Participation
- Exams/Tests
- Homework
- Projects
- Quizzes

## Open Entry/Open Exit

---

**Open Entry / Open Exit** - Not Open Entry/Open Exit

## Repeatability

---

**Is this course repeatable in accordance with Title 5 guidelines?** No

Contact Person

---

**Originator** Perry, Stephen R.