CS G131: PYTHON PROGRAMMING I

Item	Value
Curriculum Committee Approval Date	05/03/2022
Top Code	070710 - Computer Programming
Units	3 Total Units
Hours	90 Total Hours (Lecture Hours 36; Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S), • Pass/No Pass (B)

Course Description

This course will cover the fundamentals of programming using Python language. The process of software development will be discussed to include: designing, writing source code, executing, testing, and debugging. Data types, arithmetic and logical expressions, debugging, looping, branching, modularization, simple database access, simple database structures, and simple Graphical User Interface will be discussed in lectures and practiced through lab projects. Business, scientific, and mathematics applications will be designed and created. ADVISORY: CS G102. Transfer Credit: CSU; UC. C-ID: COMP 122. **C-ID:** COMP 122.

Course Level Student Learning Outcome(s)

1. Course Outcomes
2. Create programs partitioned into functions and modules.
3. Develop programming solutions using all the necessary expressions, branches, loops, functions, classes.
4. Create error handling routines.
5. Design a simple graphical user interface to satisfy the user interactions.

Course Objectives

- 1. Assess, analyze, and design software solutions for simple to moderately complex business and database problems.
- 2. Create properly documented programming solutions.
- 3. Construct the software code, mathematical formulas/expressions, and algorithms in the Python language.
- 4. Test for and eliminate coding and logic errors using debugging techniques.
- 5. Apply basic optimization techniques.
- 6. Design and implement large software solutions using manageable modules.
- 7. Develop code to interact with input, output devices and files.
- 8. Implement applications using Object Oriented Programming (OOP) paradigms through the use of classes and objects.
- 9. Summarize the evolution of programming languages illustrating how this history has led to the paradigms.

Lecture Content

Survey of Programming Languages Binary instructions and low level languages Compiled vs. translated languages Scripting languages Programming languages evolution Introduction to Computers and Programming Computer hardware Operating systems Networks and network protocols Programming languages Software libraries Input, Processing, and Output The Python programming language Abstraction and modeling Imperative programming Algorithms Data types Expressions, variables, and assignments Decision Structures and Boolean Logic The if statement The if-else statement Comparing strings Nested decision structures and the if-elif-else statement Logical operators Boolean variables Repetition Structures The while loop: a condition-controlled loop The for loop: a count-controlled loop Calculating a running total Sentinels Input validation loops Nested loops Functions and Modules Defining and calling a void function Designing a program to use functions Local variables Passing arguments to functions Global variables and global constants Files and Exceptions File input and output (I/O) Using loops to process files Processing records Exceptions Lists and Tuples Sequences Introduction to lists List slicing Finding items in lists with the in operator List methods and useful built-in functions Two-dimensional lists Text Data Basic string operations String slicing Testing, searching, and manipulating strings Formatted output Containers and Randomness Dictionaries Sets Serializing objects Class tuple Module random Classes and Object-Oriented Programming (OOP) Procedural and object-oriented programming Classes Working with instances Designing classes Inheritance Class relationships Polymorphism Recursion Introduction to recursion Problem solving with recursion Recursive algorithms Graphical User Interface (GUI) Programming Graphical user interfaces Using the tkinter module Display text with label widgets Organizing widgets with frames Button widgets and info dialog boxes Getting input with the entry widget Using labels as output fields Radio buttons and check buttons

Lab Content

Introduction to computers and Python programming Python basics More flow of control Procedural abstraction and function that return a value Functions for all subtasks I/O streams as an introduction to objects and classes Lists and tuples Strings and text processing References and dynamic arrays Defining classes Class development and modules References to lists and tuples Recursion Inheritance Exception handling Graphical User Interface (GUI) programming

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)
- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Reading Assignments

TextWebsites

Writing Assignments

Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their projects.

Out-of-class Assignments

An optional library research paper will promote further study and research in current Python programming or other related topics selected by the student and approved by the instructor.

Demonstration of Critical Thinking

Students will analyze requirements and select programming structures for efficient solution implementation. Testing and debugging will require students to perform data tracing and error isolation during program execution.

Required Writing, Problem Solving, Skills Demonstration

Students will be required to complete software development projects presented to them in the form of business automation problems requiring solution implementation. Students will be required to write documentation for their projects.

Eligible Disciplines

Computer science: Masters degree in computer science or computer engineering OR bachelors degree in either of the above AND masters degree in mathematics, cybernetics, business administration, accounting or engineering OR bachelors degree in engineering AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in mathematics AND masters degree in cybernetics, engineering mathematics, or business administration OR bachelors degree in any of the above AND a masters degree in information science, computer information systems, or information systems OR the equivalent. Note: Courses in the use of computer programs for application to a particular discipline may be classified, for the minimum qualification purposes, under the discipline of the application. Masters degree required.

Textbooks Resources

1. Required Gaddis, Tony. Starting Out with Python, 4th ed. Pearson, 2018
2. Required Perkovic, Ljubomir. Introduction to Computing Using Python, 2nd (latest) ed. Wiley, 2016 Rationale: Latest edition.