CIC Approval: 09/22/2022
BOT APPROVAL:
STATE APPROVAL:
EFFECTIVE TERM: Fall 2023

SAN DIEGO COMMUNITY COLLEGE DISTRICT
CITY, MESA, AND MIRAMAR COLLEGES
ASSOCIATE DEGREE COURSE OUTLINE

SECTION I

SUBJECT AREA AND COURSE NUMBER: Computer and Information Sciences 179

COURSE TITLE:
Introduction to Python Programming

Units:
4
Grade Only

CATALOG COURSE DESCRIPTION:

This is an introductory course in programming using the Python language and incorporating the fundamentals of object oriented programming. Topics include the use and programming of the mouse, windows, forms, menus, dialog boxes, icons, buttons, text fields, files, graphics, and other components of the Windows environment. Students learn to analyze user needs and requirements; design the user interface; assign properties to objects in the user interface; code event procedures; test and debug completed programs and applications; and complete final user documentation. This course is intended for Computer and Information Sciences majors or anyone interested in the Python programming language.

REQUISITES:

Advisory:
CISC 181 with a grade of "C" or better, or equivalent

FIELD TRIP REQUIREMENTS:
May be required

TRANSFER APPLICABILITY:
Associate Degree Credit & transfer to CSU UC Transfer Course List

CID:

TOTAL LECTURE HOURS:
48 - 54

TOTAL LAB HOURS:
48 - 54

TOTAL CONTACT HOURS:
96 - 108

OUTSIDE-OF-CLASS HOURS:
96 - 108

TOTAL STUDENT LEARNING HOURS:
192 - 216

STUDENT LEARNING OBJECTIVES:
Upon successful completion of the course the student will be able to:

1. Identify the components of the Python development environment
2. Identify basic properties of controls used in designing mobile applications, forms, and desktop interfaces
3. Use menus, forms, and dialog boxes in a Python program
4. Integrate graphics in menus and forms
5. Write, test, debug, and document Python event procedure coding
6. Determine the event procedure required to develop a Windows application after the visual interfaces have been developed
7. Use data files to save and retrieve data to a secondary storage device
8. Create and demonstrate the conversion of programs into executable and/or web-enabled applications
9. Complete procedural design documentation necessary to code Python programs
10. Systematically design and implement introductory-level Python programs from given specifications, such as problem descriptions, user stories, design criteria, input/output scenarios, functional or unit tests, pseudo code, modeling language specifications or diagrams.
11. Articulate the purpose and working of a introductory-level Python program, and its language constructs and data structures, such as through diagrams (for instance, flowcharts, Unified Modelling Language (UML) diagrams or entity-relationship (ER) diagrams), code comments, screencast videos, interactive computing notebooks (for instance, Jupyter notebooks), or written or oral communication.
12. Document external interfaces of introductory-level Python functions and methods (the application programming interface (API)), and the inner working of introductory-level Python functions and methods by in-code comments.
13. Systematically test and debug introductory-level Python programs required to validate Python programs by interactively entering arguments, utilizing functional or unit tests, running programs in a test environment, or using a debug environment (for instance, a debugger in an integrated development environment (IDE)).
14. Develop introductory-level Python programs using industry recognized tools, such as code editors, command lines (Read, execute, print loop, REPL), emulators, or integrated development environments (IDE).
15. Create a fully functional, interactive introductory-level Python program incorporating a substantial set of topics from the course.

SECTION II

1. COURSE OUTLINE AND SCOPE:

A. **Outline Of Topics:**
The following topics are included in the framework of the course but are not intended as limits on content. The order of presentation and relative emphasis will vary with each instructor.

I. Steps in the systems development life cycle for interactive Windows applications projects
   A. Planning
   B. Analysis
   C. Design
   D. Implementation
   E. Operations
   F. Support
II. Python development environment
   A. Windows environment
   B. Windows Explorer
   C. Python projects and files
   D. Intrinsic controls
III. Windows desktop forms and interfaces
   A. Controls
   B. Properties
   C. Methods
   D. Events
IV. Python event procedure coding

A. Data types, constraints, and variables
B. Program statements
C. Conditional statements
D. Loops
E. Arrays
F. Strings and typecasts

V. Elements of Python
A. Menu creation
B. Forms uses
C. Dialogue box uses

VI. Event procedures
A. Mouse
B. Keyboard
C. Menu
D. Graphic
E. Timer

VII. Procedural design documentation
A. Automatic documentation reports
B. Data flow diagrams
C. Data element dictionaries

VIII. Graphics
A. Types of graphics
B. Adding graphics to programs
C. Use of complimentary colors

IX. Data validation
A. Built-in validation events
B. Coded validation techniques

X. Data files
A. Writing data to a file
B. Reading data from a file
C. Interfacing within integrating software suites

XI. Web-enabled applications
A. Fundamentals of webpage design
B. Data display
C. Gathering data
D. Data lookups

XII. Python application program debugging
A. Manual debugging techniques
B. Automated debugging techniques
C. Optimization methods

B. **Reading Assignments:**
Reading assignments are required and may include, but are not limited to, the following:

I. Course textbook(s)
II. Articles from professional or trade journals such as Code Magazine or The Python Papers
III. Instructor handouts
IV. Articles and guides on websites related to course material such as the Microsoft Developer Network

C. **Writing Assignments:**
Writing assignments are required and may include, but are not limited to, the following:

I. Written interface form layouts with controls
II. Detailed control and event procedures
III. Program testing procedures
IV. Program operation narratives that demonstrate data flow

D. **Appropriate Outside Assignments:**
Outside assignments may include, but are not limited to, the following:

I. Researching, comparing, and analyzing interactive user interfaces in major Windows computer applications for the purpose of classroom discussion and reporting
II. Analyzing and evaluating user forms and interfaces used in Internet websites
III. Evaluating and analyzing programming requirements required in Windows applications
IV. Researching, analyzing, and evaluating Windows type interfaces and user forms utilized by the Internet, World Wide Web, and commercial service providers

E. **Appropriate Assignments that Demonstrate Critical Thinking:**
Critical thinking assignments are required and may include, but are not limited to, the following:

I. Analyzing, evaluating, and redesigning a computer user interface
II. Proposing user interface layouts
III. Creating Python programs
IV. Assessing computer applications to deductively and logically debug and correct errors
V. Recommending improvements in Windows type interfaces and associated coding

2. METHODS OF EVALUATION:

A student's grade will be based on multiple measures of performance unless the course requires no grade. Multiple measures may include, but are not limited to, the following:

I. Performance on applied out of class assignments
II. Written responses to in-class assignments
III. Responses to in-class objective and/or essay question quizzes and/or examinations
IV. Development of programs in Python
V. Demonstration of interactive one-on-one program testing and operations
VI. Participation in classroom discussion
VII. Development of program documentation

3. METHODS OF INSTRUCTION:

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

* Audio-Visual
* Computer Assisted Instruction
* Distance Education (Fully online)
* Lecture-Lab Combination
* Other (Specify)
* A. In-class, applied computer practice of concepts and techniques included in course objectives
* B. Interactive group activities including analysis, evaluation, and modification of current Windows applications

4. REQUIRED TEXTS AND SUPPLIES:
Textbooks may include, but are not limited to:

TEXTBOOKS:
1. Guzdial, Mark J. and Barbara Ericson. Introduction to Computing and Programming in Python, 4th ed. Pearson, 2016, ISBN: 9780134026343
2. Lambert, Kenneth A. Fundamentals of Python: First Programs, 2nd ed. Cengage Learning, 2018, ISBN: 9781337560092
3. Lutz, Mark. Learning Python, 5th ed. O'Reilly Media, 2013, ISBN: 9781449355739
4. Matthes, Eric. Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2nd ed. No Starch Press, 2019, ISBN: 9781593279288
5. Zelle, John. Python Programming, 3rd ed. Franklin Beedle and Assoc, 2016, ISBN: 9781590282755

MANUALS:

PERIODICALS:

SOFTWARE:

SUPPLIES:

ORIGINATOR: Allan Schougaard

CO-CONTRIBUTOR(S)
DATE: 03/15/2022