CIS 087: INTRODUCTION TO PROGRAMMING USING PYTHON

New Course Proposal

Date Submitted:Thu, 04 Oct 2018 20:52:31 GMT

Originator

fmarhuenda

Co-Contributor(s)

Name(s)
Williams, Gary

Justification / Rationale

Recommended by CIS Advisory Committee and for transfer with CSUSB IST program

Effective Term

Fall 2019

Credit Status

Credit - Degree Applicable

Subject

CIS - Computer Information Systems

Course Number

087

Full Course Title

Introduction to Programming Using Python

Short Title

INTRO TO PYTHON

Discipline

Disciplines List
Computer Information Systems (Computer network installation, microcomputer technology, computer applications)

Modality

Face-to-Face
100% Online

Catalog Description

This course provides an introduction to programming and business applications using Python. The course focuses on developing the fundamental concepts and models of application development including the basic concepts of program design, debugging, data structures, structured and object-oriented programming, problem solving, programming logic, and fundamental design techniques.

Schedule Description

This course provides an introduction to programming and business applications using Python. The course focuses on developing the fundamental concepts and models of application development including the basic concepts of program design, debugging, data structures, structured and object-oriented programming, problem solving, programming logic, and fundamental design techniques. Prerequisite: CIS 010 or Instructor approval

Lecture Units

2

Lecture Semester Hours

36

Lab Units

1

Lab Semester Hours
54

In-class Hours
90

Out-of-class Hours
72

Total Course Units
3

Total Semester Hours
162

Prerequisite Course(s)
CIS 010 or Instructor approval

Required Text and Other Instructional Materials

Resource Type
Book

Open Educational Resource
No

Formatting Style
APA

Author
Zelle, John

Title
Python Programming: An Introduction to Computer Science

Edition
3rd

City
New York

Publisher
Franklin, Beedle, and Associates

Year
2016

College Level
Yes

Flesch-Kincaid Level
12

ISBN #
978-1590282755

Class Size Maximum
32

Entrance Skills
Identify the fundamental computer concepts and terminology used for input, processing, output, and storage.

Prerequisite Course Objectives

CIS 010-Using computers effectively requires that students can express their instructions in a form that the computer program can understand and execute.
CIS 010-Students must understand what they want to accomplish, what logical steps are required to accomplish the objective, and how to submit instructions to the computer to achieve the required objective.
CIS 010-Compare and contrast the basic categories of system software and application software.
CIS 010-Explain the basic concepts and understand the uses of various categories of productivity software, including word processing, electronic spreadsheets and database management.

Entrance Skills

Identify the key features and software such as operating systems, word processors, spreadsheets, databases, communications, and graphics.

Prerequisite Course Objectives

CIS 010-Explain the basic concepts and understand the uses of various categories of productivity software, including word processing, electronic spreadsheets and database management.

Entrance Skills

Apply the principles of and solve problems with computer applications.

Prerequisite Course Objectives

CIS 010-All of the courses in the Computer Information Systems program require students to learn to think critically.
CIS 010-Students must understand what they want to accomplish, what logical steps are required to accomplish the objective, and how to submit instructions to the computer to achieve the required objective.

Course Content

1. Survey of current languages
2. Program design tools and programming environments
3. Using documentation
4. Software life-cycle including design, development, styles, documentation, testing and maintenance.
5. Programming with Numbers and Strings
6. Objects and Graphics
7. Sequences
8. Strings
9. Lists
10. Sets and Dictionaries
11. Files
12. Functions
13. Decision Structures
14. Loops
15. Simulation Design
16. Objects and Classes
17. Algorithm Design, Sorting, Searching, and Recursion
18. Procedural versus objected oriented programming

Lab Content

Lab content will be covered through individual and/or group activities. These activities are project-based and business oriented that cater to the course content.
Reinforce the following concepts:

- Program design
- Program development life-cycle
- Requirements determinants and analysis
- Modular design
- Techniques for modeling program structures
- Programming concepts

- Variables
- Literals
- Types
- Expressions
- Procedures
- Functions
- Parameters
- Operators and operations
- Decision logic
- Looping
- Sub-procedures
- Passing parameters
- Coding
- Unit testing
- Control structures

Course Objectives

	Objectives
Objective 1	Describe the basic components of the Python software development environment.
Objective 2	Define and properly use high-level programming language and control structures and syntax.
Objective 3	Describe the Python software development life cycle from concept design through documentation, testing and maintenance.
Objective 4	Define both primitive and compound data types and give examples in Python of each type.
Objective 5	Use Python variable expressions in a program to compute numeric and string results.
Objective 6	Explain what an algorithm is and give examples of how algorithms are implemented in a Python program.
Objective 7	Design, implement, test, and debug a program or function that can be used in programs, and demonstrate the way parameters are passed in such functions.
Objective 8	Write Python programs using object-oriented design, and contrast the difference between object-oriented and procedural code.
Objective 9	Describe the internal representation of characters, strings, records, and arrays.

Student Learning Outcomes

	Upon satisfactory completion of this course, students will be able to:
Outcome 1	Compose and debug Python programs which make use of the fundamental control structures and function-building techniques common to all programming languages.
Outcome 2	Utilize object-oriented programming techniques to design and implement a clear, well-structured Python program using design classes and objects.
Outcome 3	Demonstrate basic problem solving skills by analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language.

Methods of Instruction

Method	Please provide a description or examples of how each instructional method will be used in this course.
Lecture	Presentation of class lectures/discussions/demonstrations to model and explain the fundamental concepts of programming, algorithms, and problem solving.
Laboratory	Students will create Python programs to solve real-life business needs.
Demonstration, Repetition/Practice	Students will have to incorporate instructor feedback on each of their Python lab activities and resubmit for grading.
Technology-based instruction	Use of Integrated Development Environment (IDE) for software development.

Activity	Activities focused on addressing areas of improvement in the fundamental concepts of Object-Oriented programming, such as elevator programming and algorithm flow.
Discussion	Students will discuss assigned Business/CIS cases.
Collaborative/Team	Work group consisting of developing an algorithm to fulfill business needs.
Participation	Students will participate in discussion regarding best practices in object-oriented programming.

Methods of Evaluation

Method	Please provide a description or examples of how each evaluation method will be used in this course.	Type of Assignment
Written homework	Write reports of system development procedures. Approximately 15 hours outside of class.	Out of Class Only
Group activity participation/observation	Work group consisting of developing an algorithm to fulfill business needs.	In Class Only
Product/project development evaluation	Develop Python programs based on group work algorithm.	In Class Only
Mid-term and final evaluations	Students will complete quizzes and exams throughout the course.	In Class Only
Reading reports	Read Python documentation. Approximately 15 hours	Out of Class Only
Student participation/contribution	Students will participate in discussions regarding assigned Business/CIS cases.	In Class Only
Laboratory projects	Integrated Development Learning Environment (IDLE) lab simulations. Build Python programs.	In Class Only
Presentations/student demonstration observations	Students will demonstrate to the class the function of each line of their Python source code.	In Class Only
Computational/problem-solving evaluations	Students will be required to create a Python program that solves a problem. Approximately 20 hours.	Out of Class Only
Product/project development evaluation	Students will evaluate presented code and provide an efficiency report. Approximately 15 hours.	In and Out of Class
True/false/multiple choice examinations	The course will contain various types of examinations. Approximately 10 hours.	In and Out of Class

Assignments

Other In-class Assignments

- Create a Python program
- Design user interface using controls (text box, label, button, menu . . .)
- Write event procedures to handle events from controls
- Use IF statements or Case structure to make decision based on the test of a condition
- Break down a complicated problem (calculation) into smaller units. Solve each unit using a procedure or function
- Create multiform project
- Use loop structure to repeat an operation multiple times.
- Create console application that calculates Gross Pay, Net Pay, and Taxes Withheld based on specified user input.
- Create console application that utilizes selection control to create the ‘Menu Driven Banking’ application.

Other Out-of-class Assignments

Readings: Textbooks, Supporting references, Websites, Periodicals, course handouts.
Students will be asked to read white papers and Internet based information describing algorithms in order to discuss the topic of algorithm creation during class.

Writing: Analyses, Reports, Responses/Reactions, Critiques.
Students will be asked to write a detailed list of pseudocode for describing a simple process and evaluating how that would translate to the logic of object oriented programming.

Critical Thinking: Summarizes the problem/question/work assignment, considers the influence of context and assumptions, communicates own perspective or position, Identifies implications and consequences, applies appropriate tools in problem-solving. Students will be asked to compare and contrast popular programming languages
and determine which language is best used for which application, defending their
decisions.

Grade Methods

Letter Grade Only

Distance Education Checklist

Lab Courses

How will the lab component of your course be differentiated from the lecture component of the course?

The lecture portion of the course will consist of readings and activities (e.g. pseudocode, algorithms) that will build knowledge of the Python programming language and program structure.
Students will be writing Python scripts and programs during the lab component of the course.

From the COR list, what activities are specified as lab, and how will those be monitored by the instructor?

Activities that involve the students writing Python scripts will be specified as lab. The course will contain Canvas weekly assignments that require students complete working Python programs that achieve a particular goal.

How will you assess the online delivery of lab activities?

Lab activities will require that students install an Integrated Development Environment (IDE) to create their software; this has been the primary issue in all our other programming courses. We have developed a robust module centered around the IDE so students can complete their lab activities.
We realized that IDE was the issue because fewer students turned in their lab coding assignments. We will use these type of techniques to assess our delivery methods.

Instructional Materials and Resources

If you use any other technologies in addition to the college LMS, what other technologies will you use and how are you ensuring student data security?

We will be using IDEs that is open source to develop code. It will not contain or traffic student data.

If used, explain how specific materials and resources outside the LMS will be used to enhance student learning.

The IDE is necessary for the development of all computer programs.

Effective Student/Faculty Contact

Which of the following methods of regular, timely, and effective student/faculty contact will be used in this course?

Within Course Management System:

Timely feedback and return of student work as specified in the syllabus
Discussion forums with substantive instructor participation
Regular virtual office hours
Private messages
Online quizzes and examinations
Video or audio feedback
Weekly announcements

External to Course Management System:

Direct e-mail
E-portfolios/blogs/wikis
Posted audio/video (including YouTube, 3cmediasolutions, etc.)
Synchronous audio/video
Teleconferencing
Telephone contact/voicemail

Briefly discuss how the selected strategies above will be used to maintain Regular Effective Contact in the course.

There will be weekly discussions regarding topics related to the course with appropriate instructor participation.
Students will create and upload Python programs of their creation. These activities will receive appropriate instructor feedback.

Online Course Enrollment

Maximum enrollment for online sections of this course
32

Other Information

Provide any other relevant information that will help the Curriculum Committee assess the viability of offering this course in an online or hybrid modality.
This course is really meant as a sister course to our CIS 86 (Visual Basic) course but with a different language. We have been successfully delivering CIS 86 online since the inception of COD's online program.

Comparable Transfer Course Information

University System
UC
Campus
UC Santa Cruz

Course Number
CMPS 5P
Course Title
Introduction to Programming in Python

Catalog Year
2018

University System
CSU
Campus
CSU San Bernardino

Course Number
IST 282
Course Title
Business Systems 1

Catalog Year
2018

MIS Course Data

CIP Code
11.0201 - Computer Programming/Programmer, General.

TOP Code
070710 - Computer Programming

SAM Code
B - Advanced Occupational

Basic Skills Status
Not Basic Skills

Prior College Level
Not applicable

Cooperative Work Experience
Not a Coop Course

Course Classification Status
Credit Course

Approved Special Class
Not special class

Noncredit Category
Not Applicable, Credit Course

Funding Agency Category
Not Applicable

Program Status
Program Applicable

Transfer Status
Transferable to UC & CSU

Allow Audit
No

Repeatability
No

Materials Fee
No

Additional Fees?
No

Files Uploaded

Attach relevant documents (example: Advisory Committee or Department Minutes)
Advisory Meeting_Minutes Spring 2017.docx

Approvals

Curriculum Committee Approval Date
11/6/2018

Academic Senate Approval Date
11/29/2018

Board of Trustees Approval Date
12/14/2018

Chancellor's Office Approval Date
1/07/2019

Course Control Number
CCC000599876

Programs referencing this course
Computer Information Systems Associate of Science and Transfer Preparation (http://catalog.collegeofthedesert.eduundefined?key=221)
Computer Information Systems AS Degree for Employment Preparation (http://catalog.collegeofthedesert.eduundefined?key=61)