Cerro Coso College

Course Outline of Record Report

10/12/2021

ITC251 : Introduction to Programming Concepts and Methodologies

General Information

Author:	• Matthew Hightower • Karnes, Valerie
Course Code (CB01) :	ITC251
Course Title (CB02) :	Introduction to Programming Concepts and Methodologies
Department:	Business Information Technolog
Proposal Start:	Spring 2019
TOP Code (CB03) :	(0702.00) Computer Information Systems
SAM Code (CB09) :	Clearly Occupational
Distance Education Approved:	Yes
Course Control Number (CB00) :	CCC000564176
Curriculum Committee Approval Date:	03/16/2018
Board of Trustees Approval Date:	06/14/2018
External Review Approval Date:	Pending
Course Description:	This course is an introduction to the fundamental concepts and models of application development including the basic concepts of program design, data structures, programming, problem solving, programming logic, and fundamental design techniques for event-driven programs. Students receive hands-on experience with a modern application programming language and development platform. Note: This course was formerly CSCI C251.
Submission Type:	Improvement to Program of Study Per program review, change CSCI to IT designation for program clarification and SLO data assessment.
Author:	No value

Faculty Minimum Qualifications

Master Discipline Preferred:	• Computer Information Systems (Computer network installation, microcomputer technology, computer applications) • Computer Science
Alternate Master Discipline Preferred:	• Computer Science
Bachelors or Associates Discipline Preferred:	• Computer Information Systems (Computer network installation, microcomputer technology, computer applications)
Additional Bachelors or Associates Discipline Preferred:	No value

Course Development Options

Basic Skills Status (CB08)

Course is not a basic skills course.

☐ Allow Students to Gain Credit by Exam/Challenge

Rationale For Credit By Exam/Challenge

No value

Course Support Course Status (CB26)

No value

Course Special Class Status (CB13)

Course is not a special class.

Allowed Number of Retakes

0

Retake Policy Description

Type:|Non-Repeatable Credit

Grade Options

- Letter Grade Methods
- Pass/No Pass

Course Prior To College Level (CB21)

Not applicable.

☑ Allow Students To Audit Course

Associated Programs

☑ Course is part of a program (CB24)

Associated Program	Award Type	Active
Cyber Security Technology	A.S. Degree Major	Spring 2018
CC Liberal Arts: Mathematics & Science	A.A. Degree Major	Summer 2018 to Fall 2020
Associate in Science Degree In Mathematics for Transfer	A.A. Degree for Transfer	Summer 2018
Cyber Security Technician	Certificate of Achievement	Spring 2018
CC Computer Information Systems-	Certificate of Achievement	Spring 2018 to Summer 2019
CC Computer Information Systems	A.S. Degree Major	Spring 2018 to Summer 2019
CC Information Technology	Certificate of Achievement	Summer 2019
CC Information Technology	A.S. Degree Major	Summer 2019
Cloud Computing	Certificate of Achievement	Fall 2020 to Spring 2021

Liberal Arts: Mathematics & Science Associate in Arts Degree	A.A. Degree Major	Fall 2020
Web Professional Associate of Science (In Development)	A.S. Degree Major	Fall 2022
Web Professional Certificate of Achievement (In Development)	Certificate of Achievement	Fall 2022

Transferability & Gen. Ed. Options

Course General Education Status (CB25)

No value

Transferability

Transferable to both UC and CSU

Transferability Status

Approved

Units and Hours

Summary

Minimum Credit Units (CB07)	3
Maximum Credit Units (CB06)	3
Total Course In-Class (Contact) Hours	90
Total Course Out-of-Class Hours	72
Total Student Learning Hours	162
Faculty Load	0

Credit / Non-Credit Options

Course Credit Status (CB04)

Credit - Degree Applicable

Course Non Credit Category (CB22)

Credit Course.

Non-Credit Characteristic

No Value

Course Classification Status (CB11)

Credit Course.

Funding Agency Category (CB23)

Not Applicable.

☐ Cooperative Work Experience Education Status (CB10)

☐ Variable Credit Course

Weekly Student Hours

	In Class	Out of Classs
Lecture Hours	2	4

Course Student Hours

Course Duration (Weeks)	18
Hours per unit divisor	54

Laboratory Hours	3	0
Activity Hours	0	0

Course In-Class (Contact) Hours	
Lecture	36
Laboratory	54
Activity	0
Total	90

Course Out-of-Class Hours	
Lecture	72
Laboratory	0
Activity	0
Total	72

Time Commitment Notes for Students

No value

Faculty Load

Extra Duties: 0

Faculty Load: 0

Units and Hours - Weekly Specialty Hours

Activity Name	Type	In Class	Out of Class
No Value	No Value	No Value	No Value

Pre-requisites, Co-requisites, Anti-requisites and Advisories

Prerequisite

ITC101 - Introduction to Computer Information Systems

IT C101 Course Prerequisite for Conditions of Enrollment for IT 251
Students are expected to have a working knowledge of applications programs and file management (example: Students should know how to work with Windows applications such as Word and Excel and should know how to save, rename, find, and delete files). In addition, students should have an introductory knowledge of how computers and information systems are used in business, how computers are programmed, and how to install software (example: Students will need to install the programming language and Integrated Development Environment (IDE) and the basics of how computer programs work and what they are used for).

Entrance Skills

Entrance Skills	Description

No value	No value

Limitations on Enrollment

Limitations on Enrollment	Description
No value	No value

Specifications

Methods of Instruction

Methods of Instruction	Other
Rationale	Other Methods: A. PowerPoint lectures demonstrating the logic, syntax and use of programming controls, properties, structures, and classes.
Methods of Instruction	Written work
Rationale	No value
Methods of Instruction	Project-based learning
Rationale	No value
Methods of Instruction	Outside reading
Rationale	No value
Methods of Instruction	Laboratory
Rationale	No value
Methods of Instruction	Lecture
Rationale	No value
Methods of Instruction	Demonstration
Rationale	No value

Methods of Instruction	Discussion
Rationale	No value

Methods of Instruction	Instruction through examination or quizzing
Rationale	No value

Methods of Instruction	Computational Work
Rationale	No value

Assignments

A. Chapter reading (Example: Reading the assigned chapters from the textbook based on the topics for the week).
B. Weekly step-by-step coding assignments (Example - Follow and interpret examples to write and test program coding assignments).
C. Weekly application design assignments (Example:Design program flow and control for a variety of programming tasks based on requirements received).

Methods of Evaluation	Rationale
Final Exam	Final Exam demonstrating comprehensive mastery of material presented. Example: Multiple choice and essay question exam covering all concepts of the course.
Participation	Weekly discussion participation demonstrating understanding of programming concepts . Example: Please use the forum to post your progress and your thoughts regarding the Chapter assignments. Also post any questions or problems that you may be having with the assignment. Do this by Thursday night. Reply to at least two other posts between Friday and Sunday. You can either assist another student by providing help with their questions or respond to their posted thoughts where appropriate..
Homework	Weekly programming assignments demonstrating mastery of new programming material. Example: Program coding assignment on use of control structures.
Tests	Midterm Exam(s) demonstrating mastery of material in the first half of instruction. Example: Multiple choice and essay question exam covering variables, types, expressions, and methods.

Equipment

Students need access to a computer (Windows, Mac, or Linux) with the Python programming environment.

Textbooks

Author	Title	Publisher	Date	ISBN
Horstmann, C. & Necaise, R.	Python for Everyone (2nd ed.)	Wiley	2016	978-1-119-05655-3

Other Instructional Materials

Description	Python 3.5+ programming environment. This software is free and is available for Windows, Mac OS, and Linux environments.
Author	No value
Citation	No value

Materials Fee

No

Learning Outcomes and Objectives

Course Objectives

Use primitive data types and data structures offered by the development environment.

Choose an appropriate data structure for modeling a simple problem.

Identify basic programming concepts.

Write simple applications that relate to a specific domain.

Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.

Test applications with sample data.

Apply core program control structures.

CSLOs

Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and functions. Expected SLO Performance: 70.0

ISLOs Core ISLOs	Students who are completing a program will be able to think critically and creatively and apply reasoning.

Test applications with sample data. Expected SLO Performance: 70.0

Apply core program control and data structures. Expected SLO Performance: 70.0

Outline

Course Outline

A.Programming Overview
- a.History of programming
- b.Types of languages
  - i.Procedural
  - ii.Event-driven
  - iii.Object oriented

B.Program design
- a.Program development lifecycle
- b.Requirements determinants and analysis
- c.Modular design
- d.Techniques for modeling program structures

C.Programming concepts
- a.Variables
- b.Literals
- c.Types
- d.Expressions

D.Control Structures
- a.Operators and operations
- b.Decision logic
  - i.If then
  - ii.If then else
- c.Looping
  - i.For Next
  - ii.Do Loop
  - iii.Counters
  - iv.Accumulators

E.Procedures
- a.Functions
- b.Sub-procedures
- c.Passing parameters

F.Coding
- a.Unit testing
- b.Debugging tools
- c.Implementation

G. File and array processing
- a. Arrays and matrices
- b. Sequential files
- c. Database files

Lab Outline

Students complete step-by-step programming examples to introduce programming concepts. Building upon the examples; students design and develop Graphical User Interfaces (GUIs) weekly.

A. Introduction to Windows-based Applications
- a. Working with Controls
- b. Creating a GUI

B. Writing Algorithms
- a. Pseudocode/Flowcharting
- b. Coding; Testing; Debugging

C. Variables and Constants
- a. Creating variables and named constants
- b. Using variables and constants
- c. Arithmetic work with variables

D. Selection Structures
- a. Comparison and Relational Operators
- b. If
- c. If/Else

d. Nested
e. Case Selection

E. Repetition Structures
a. For Next
b. Do Loop
c. Counters and Accumulators
d. Nesting

F. Sub and Function Procedures
a. Creating sub and function procedures
b. Using sub and function procedures
c. Passing information
d. Passing variables
e. Return variables

G. Arrays
a. One-dimensional arrays
b. Parallel one-dimensional arrays
c. Two-dimensional arrays

I. Structures and Sequential Access Files
a. Create a structure
b. Sequential access files
c. Reading and writing to file

Delivery Methods and Distance Education

Delivery Method: Please list all that apply -Face to face -Online (purely online no face-to-face contact) -Online with some required face-to-face meetings ("Hybrid") -Online course with on ground testing -iTV – Interactive video = Face to face course with significant required activities in a distance modality -Other

Face 2 Face
Online
Hybrid
Interactive

Rigor Statement: Assignments and evaluations should be of the same rigor as those used in the on-ground course. If they are not the same as those noted in the COR on the Methods of Evaluation and out-of-class assignments pages, indicate what the differences are and why they are being used. For instance, if labs, field trips, or site visits are required in the face to face section of this course, how will these requirements be met with the same rigor in the Distance Education section?

All assignments in distance education courses (online, hybrid and iTV) of IT C251 are of the same rigor as those in the on-ground course, except that students in purely online sections will submit all of their assignments virtually. Instructor evaluation of student work in distance education courses is the same as in the on-ground course, except that evaluation of student work in online is presented virtually. Instead of onsite lectures, hybrid and online courses use a variety of methods including, but not limited to videos, interactive simulations and written lecture notes.

Effective Student-Instructor Contact: Good practice requires both asynchronous and synchronous contact for effective contact. List the methods expected of all instructors teaching the course. -Learning Management System -Discussion Forums -Moodle Message -Other Contact -Chat/Instant Messaging -E-mail -Face-to-face meeting(s) -Newsgroup/Discussion Board -Proctored Exam -Telephone -iTV -Interactive Video -Other (specify)

Learning Management System
Discussion Forums
Inbox and/or email

Software and Equipment: What additional software or hardware, if any, is required for this course purely because of its delivery mode? How is technical support to be provided?

No Value

Accessibility: Section 508 of the Rehabilitation Act requires access to the Federal government's electronic and information technology. The law covers all types of electronic and information technology in the Federal sector and is not limited to assistive technologies used by people with disabilities. It applies to all Federal agencies when they develop, procure, maintain, or use such technology. Federal agencies must ensure that this technology is accessible to employees and the public to the extent it does not pose an "undue burden". I am using -iTV—Interactive Video only -Learning management system -Publisher course with learning management system interface.

s508_itv
s508_publisher

Class Size: Good practice is that section size should be no greater in distance ed modes than in regular face-to-face versions of the course. Will the recommended section size be lower than in on-ground sections? If so, explain why.

The size of the online section is not constrained by computer lab seating. Historical enrollment patterns have shown that the face to face and online section sizes are similar in number of students.