Cerro Coso College

Course Outline of Record Report

10/13/2021

CSCIC265 : Introductory C++ Programming

General Information

Author:	-
Course Code (CB01) :	CSCIC265
Course Title (CB02) :	Introductory C++ Programming
Department:	Business Information Technolog
Proposal Start:	Fall 2013
TOP Code (CB03) :	(0706.00) Computer Science (transfer)
SAM Code (CB09) :	Clearly Occupational
Distance Education Approved:	Yes
Course Control Number (CB00) :	CCC000291874
Curriculum Committee Approval Date:	10/04/2013
Board of Trustees Approval Date:	11/14/2013
External Review Approval Date:	02/25/2014
Course Description:	This course is an introduction to C++ object-oriented programming, including fundamentals, logic, algorithm development, classes, functions and inheritance.
Submission Type:	New Course
Author:	No value

Faculty Minimum Qualifications

Master Discipline Preferred:	• Computer Science
Alternate Master Discipline Preferred:	No value
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.

Course Special Class Status (CB13)

Course is not a special class.

Grade Options

- Letter Grade Methods
- Satisfactory Progress

☐ Allow Students to Gain Credit by Exam/Challenge

Allowed Number of Retakes

0

Course Prior To College Level (CB21)

Not applicable.

Rationale For Credit By Exam/Challenge

Retake Policy Description

✓ Allow Students To Audit C

No value	Type:|Non-Repeatable Credit	Allow Students To Audit Course

Course Support Course Status (CB26)

No value

Associated Programs

✓ Course is part of a program (CB24)

Associated Program	Award Type	Active
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
Liberal Arts: Mathematics & Science Associate in Arts Degree	A.A. Degree Major	Fall 2020

Transferability & Gen. Ed. Options

Course General Education Status (CB25)

No value

Transferability	Transferability Status
Transferable to both UC and CSU	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)	**Course Non Credit Category (CB22)**	**Non-Credit Characteristic**
Credit - Degree Applicable	Credit Course.	No Value
Course Classification Status (CB11)	**Funding Agency Category (CB23)**	☐ Cooperative Work Experience Education Status (CB10)
Credit Course.	Not Applicable.	

☐ Variable Credit Course

Weekly Student Hours

	In Class	Out of Classs
Lecture Hours	2	4
Laboratory Hours	3	0
Activity Hours	0	0

Course Student Hours

Course Duration (Weeks)	18
Hours per unit divisor	0
Course In-Class (Contact) Hours	
Lecture	0
Laboratory	0
Activity	0
Total	90
Course Out-of-Class Hours	
Lecture	0
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

CSCIC252 - Introduction to Computer Science

Basic problem solving and programming skills such as algorithms development and variable rules allowing students to progress to the more advanced skills they will need in the workforce.

AND

Prerequisite

CSCIC251 - Introduction to Programming Concepts and Methodologies

Basic problem solving and programming skills such as algorithms development and variable rules allowing students to progress to the more advanced skills they will need in the workforce.

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. Lectures demonstrating the logic, syntax and use of C++ programming controls, properties, structures, and classes.
Methods of Instruction	Outside reading
Rationale	No value
Methods of Instruction	Problem Solving

Rationale	No value
Methods of Instruction	Skills Development and Performance
Rationale	No value
Methods of Instruction	Lecture
Rationale	No value
Methods of Instruction	Laboratory
Rationale	No value
Methods of Instruction	Demonstration
Rationale	No value

Assignments

A. Reading Text - Preparing for class by reading the chapters assigned
B. Programming Assignments - Programming assignments every week
C. Homework Assignments - Problem sets as handouts or from the text to practice concepts.

Methods of Evaluation	Rationale
Participation	Discussion Participation demonstrating understanding of C++ concepts
Homework	Weekly lab work demonstrating understanding of new material presented
Tests	Exams demonstrating mastery of material in the instruction
Homework	Weekly Programming Assignments demonstrating mastery of new programming material

Equipment

No Value

Textbooks

Author	Title	Publisher	Date	ISBN
	Deitel, P., Deitel, H.. (2014) C++ How to Program (Early Objects Version), 9/E, 9th, Prentic Hall			

Other Instructional Materials

Description	Software: Microsoft. Microsoft Visual Studio C++ Express Edition 2010, 2010 ed. -C++ Integrated Development Environment
Author	
Citation	Introductory C++ Programming

Materials Fee

No

Learning Outcomes and Objectives

Course Objectives

No value

CSLOs

Define and apply the fundamentals, structure, logic and syntax of C++ programming. Expected SLO Performance: 70.0

Identify the terminology associated with object-oriented programming and C++. Expected SLO Performance: 70.0

Develop, design and code simple to moderate applications using C++ Expected SLO Performance: 70.0

Science Liberal Arts: Mathematics & Science AA Degree	Apply algebraic, graphical, numerical, and other methods to solve applied problems in the areas of mathematics, natural sciences, computer graphics, and computer animation.

Analyze program code. Expected SLO Performance: 70.0

Interpret and use strings, variables, repetition structures, pointers, arrays, structures, functions, friends, inheritance, classes and objects. Expected SLO Performance: 70.0

Identify memory management principles and explain how they affect the design and implementation of C++ programs. Expected SLO Performance: 70.0

Outline

Course Outline

A. Overview
a. Overview of programming
b. History of programming languages
c. Object-Oriented (OO) Terminology
B. Introduction to C++ Programming
a. Programming universals
b. Introduce variables
c. Introduce the const qualifier
d. Create comments
e. Introduce cout and cin

C. Evaluating C++ Expressions
a. Binary arithmetic operators
b. Arithmetic operations &ndash: precedence and associative property
c. Unary arithmetic operators
d. Boolean expressions
D. Making Decisions
a. If; If-else statements
b. Nested If statements
c. Switch statement
d. Conditional operators
e. Logical operators
f. Common errors
E. Performing Loops
a. While loop
b. Accumulators
c. For loop
d. Nested loops
e. Common errors
F. Arrays; Strings and Pointers
a. Memory addresses
b. Use of arrays
c. Common errors
d. Parallel arrays
e. String handling
f. Pointers
G. C++ Functions
a. Simple functions in files
b. Procedural abstraction
c. Scope
d. Pass and return values
e. Common errors
f. Objects as arguments
g. Passing addresses
h. Reference variables
i. Passing arrays
j. Inline
k. Default parameters
l. Overload
H. Classes
a. Create class
b. Encapsulating
c. Implementation
d. Public and private
e. Scope
f. Static class members
g. this pointer
h. Polymorphism
I. Class Features and Design Issues
a. Roles of member functions
b. Constructors with or without arguments
c. Overload constructors
d. Create destructors
e. Composition
f. Preprocessor directives
g. Mastering techniques
J. Friends and Overloading Operators
a. Basics and declaration of friend
b. Accessing data from two classes
c. Rules of operator overloading
d. Overload insertion; extraction; prefix; postfix; ==; =; subscript
and parenthesis operators
K. Inheritance
a. Benefits of inheritance
b. Derived class
c. Restrictions
d. Class access specifier
e. Overrides

f. Base class constructor
g. Multiple inheritance
h. Problems posed with multiple inheritance
L. Templates
a. Structure of function templates
b. Overloading function templates
c. Create function templates
d. Class templates
e. Container classes
M. Handling Exceptions
a. Throw and catch exceptions
b. try blocks
c. Default exception handler
d. Exception specifications
e. Stacks
f. Memory-allocation exceptions
N. Advanced Input and Output
a. cin and cout
b. istream
c. ostream
d. Manipulators
e. Data hierarchy
f. Read and write objects

Lab Outline

A. Introduction to C++ Programming
a. Variables
b. Cin and cout
B. Evaluating C++ Expressions
a. Binary arithmetic
b. Program C++ arithmetic
C. Making Decisions
a. If; If-else statements
b. Nested If statements
c. Switch statement
d. Conditional operators
e. Logical operators
f. Common errors
D. Performing Loops
a. While loop
b. For loop
c. Nested loops
d. Common errors
E. Arrays; Strings and Pointers
a. Use arrays
b. Common errors
c. Parallel arrays
d. String handling
e. Pointers
F. C++ Functions
a. Simple functions
b. Pass and return values
c. Common errors
d. Reference variables
e. Overload
G. Classes
a. Create class
b. Static class members
c. this pointer
H. Class Features and Design Issues
a. Constructors with or without arguments
b. Create destructors
I. Friends and Overloading Operators

a. Basics and declaration of friend
b. Overload insertion; extraction; prefix; postfix; ==; =; subscript and parenthesis operators
J. Inheritance
a. Create inheritance hiearachy
b. Create multiple inheritance programs
K. Templates
a. Create function templates
b. Class templates
c. Container classes
L. Handling Exceptions
a. Throw and catch exceptions
b. try blocks
M. Advanced Input and Output
a. istream
b. ostream
c. Manipulators

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

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?

Lab assignments available through MyProgrammingLab with the textbook.

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)

contact_moodle_forums
contact_moodle_message
contact_chat
contact_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?

Visual Studio C++ Express Edition 2010 (or another C++ compiler)

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_moodle
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.

No Value