

Glendale College

Course Outline of Record Report

Course ID 005204
Revision - March 2025

CS/IS137 : C++ and Advanced Topics

General Information

Author:	<ul style="list-style-type: none"> Edwin Sookiassian Biehl, Tony
Attachments:	DE Addendum_CSIS_137 COR_09:01:2020 CoDE_05:23:2023.pdf DE Addendum_CSIS_137 COR_09_01_2020 CoDE_05_23_2023.pdf
Course Code (CB01) :	CS/IS137
Course Title (CB02) :	C++ and Advanced Topics
Department:	CSIS
Proposal Start:	Spring 2026
TOP Code (CB03) :	(0707.10) Computer Programming
CIP Code:	(11.0201) Computer Programming/Programmer, General.
SAM Code (CB09) :	B - Advanced Occupational
Distance Education Approved:	No
Will this course be taught asynchronously?:	Yes
Course Control Number (CB00) :	CCC000579619
Curriculum Committee Approval Date:	03/26/2025
Board of Trustees Approval Date:	06/17/2025
Last Cyclical Review Date:	03/26/2025
Course Description and Course Note:	CS/IS 137 prepares students for C++ programming in the workplace and other real-world environments. Students learn advanced object-oriented programming concepts and other advanced concepts such as templates and generics, files and streams, and operator overloading. Students also choose advanced projects.
Justification:	Mandatory Revision
Academic Career:	<ul style="list-style-type: none"> Credit
Mode of Delivery:	No value
Author:	<ul style="list-style-type: none"> Edwin Sookiassian Biehl, Tony
Course Family:	No value

Academic Senate Discipline

Primary Discipline:	<ul style="list-style-type: none"> Computer Science
Alternate Discipline:	No value
Alternate Discipline:	No value

Course Development

Basic Skill Status (CB08)

Course is not a basic skills course.

Allow Students to Gain Credit by Exam/Challenge

Course Special Class Status (CB13)

Course is not a special class.

Pre-Collegiate Level (CB21)

Not applicable.

Grading Basis

- Grade with Pass / No-Pass Option

Course Support Course Status (CB26)

Course is not a support course

General Education and C-ID

General Education Status (CB25)

Not Applicable

Transferability

Transferable to both UC and CSU

Transferability Status

Approved

Units and Hours

Summary

Minimum Credit Units (CB07)	4
Maximum Credit Units (CB06)	4
Total Course In-Class (Contact) Hours	72
Total Course Out-of-Class Hours	144
Total Student Learning Hours	216

Credit / Non-Credit Options

Course Type (CB04)

Credit - Degree Applicable

Noncredit Course Category (CB22)

Credit Course.

Noncredit Special Characteristics

No Value

Course Classification Code (CB11)

Credit Course.

Variable Credit Course

Funding Agency Category (CB23)

Not Applicable.

Cooperative Work Experience Education

Status (CB10)

Weekly Student Hours

	In Class	Out of Class
Lecture Hours	4	8
Laboratory Hours	0	0
Studio Hours	0	0

Course Student Hours

Course Duration (Weeks)	18
Hours per unit divisor	0
Course In-Class (Contact) Hours	
Lecture	72

Laboratory	0
Studio	0
Total	72
Course Out-of-Class Hours	
Lecture	144
Laboratory	0
Studio	0
Total	144

Time Commitment Notes for Students

No value

Units and Hours - Weekly Specialty Hours

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

Prerequisites, Corequisites, Recommended Corequisites, and Recommended Preparation

Prerequisite

CS/IS135 - Programming In C/C++

Objectives

- Demonstrate an understanding of object-oriented programming concepts and object-oriented design in creating a program.

Entry Standards

Entry Standards	Description
Design, code, and debug basic object-based programs.	No Value

Course Limitations

Cross Listed or Equivalent Course	Description
-----------------------------------	-------------

No value	No value
----------	----------

Specifications

Methods of Instruction

Methods of Instruction	Lecture
------------------------	---------

Methods of Instruction	Demonstrations
------------------------	----------------

Methods of Instruction	Presentations
------------------------	---------------

- Out of Class Assignments**
- Group and individual programming projects (e.g. move a sprite under program and algorithmic control within a window)

Methods of Evaluation	Description of Activity/Interaction
Exam/Quiz/Test	Final examination
Exam/Quiz/Test	Quizzes
Exam/Quiz/Test	Midterm examinations

Textbook Rationale
No Value

Textbooks

Author	Title	Publisher	Date	ISBN
Deitel, Paul	C++:How to Program	Upper Saddle River: Prentice Hall	2023	9780138101640

Other Instructional Materials (i.e. OER, handouts)
No Value

Learning Outcomes

Course Objectives

Demonstrate understanding of object oriented programming concepts and the application of those concepts to the C++ language.

Develop an advanced understanding of techniques used in the areas of objected oriented programming.

Create software development projects.

Apply advanced features such as templates and the standard template library.

SLOs

- Apply advanced programming techniques involving file streams, event-driven programming, and OS interfaces. Expected Outcome Performance: 70.0
- Describe the practical uses of object oriented programing. Expected Outcome Performance: 70.0
- Apply programming skills to effectively solve practical problems encountered in software development. Expected Outcome Performance: 70.0

Additional SLO Information

Does this proposal include revisions that might improve student attainment of course learning outcomes?

No

Is this proposal submitted in response to learning outcomes assessment data?

No

If yes was selected in either of the above questions for learning outcomes, explain and attach evidence of discussions about learning outcomes.

No Value

SLO Evidence

No Value

Course Content

Lecture Content

Introduction to Visual Studio (3 hours)

- Using the interface
- Creating a basic window
- Background and basics of using .NET and Visual C++

Object Oriented Programming – A Deeper Look (8 hours)

- Const (constant) objects and const member functions
- Composition: objects as members of classes
- Friend functions and friend classes
- Dynamic memory management with operators new and delete
- Static class members
- Data abstraction and information hiding

Operator Overloading; String and Array Objects (8 hours)

Object Oriented Programming – Inheritance (14 hours)

- Base classes and derived classes
- Protected members
- Relationship between base classes and derived classes
- Constructors and destructors in derived classes

Object Oriented Programming – Polymorphism (14 hours)

- Fundamentals and examples
- Virtual functions

Templates and Generics (8 hours)

- Creating user defined templates
- Using generics

Files and Streams (8 hours)

- File handling mechanisms
- Passing streams to functions

Standard Template Library (7 hours)

- Overview of pre-defined data types (ADT)
- Use of those ADT

Basics of GUI Programming (2 hours)

Total Hours: 72

Additional Information

Repeatability

Not Repeatable

Justification (if repeatable was chosen above)

No Value

Is it possible this course will have a material fee?

No Value

I have contacted my library liaison (<https://campusguides.glendale.edu/faculty/liasons>):

No Value

What term(s) will this course be offered?

Fall/Spring

Will any additional resources be needed for this course? (Click all that apply)

No Value

If additional resources are needed, add a brief description and cost in the box provided.

No Value

Resources

Did you contact your departmental library liaison?

No

If yes, who is your departmental library liason?

No Value

Did you contact the DEIA liaison?

No

Were there any DEIA changes made to this outline?

No Value

If yes, in what areas were these changes made:

No Value

Will any additional resources be needed for this course? (Click all that apply)

- No

If additional resources are needed, add a brief description and cost in the box provided.

No Value