

# CSCI 0066 - OBJECT-ORIENTED PROGRAMMING USING C++

---

## Catalog Description

Prerequisite: Completion of CSCI 12 with grade of "C" or better

Advisory: Completion of CSCI 46 with grade of "C" or better

Hours: 72 (54 lecture, 18 laboratory)

Description: An introduction to the concepts of object-oriented programming and the application of the C++ language. Extensive programming practice using C++ as the vehicle toward modular, reusable object-oriented code. (CSU, UC)

## Course Student Learning Outcomes

- CSLO #1: Apply C++ concepts including structures, classes, methods, objects, and primitive types, referencing basic C++ libraries.
- CSLO #2: Use the object-oriented language concepts available in C++ to demonstrate encapsulation, abstraction, instantiation, overloading, inheritance, composition, and polymorphism.
- CSLO #3: Apply the object-oriented language concepts available in C++ to design and implement a program that applies to a real-world application.

## Effective Term

Fall 2023

## Course Type

Credit - Degree-applicable

## Contact Hours

72

## Outside of Class Hours

90

## Total Student Learning Hours

162

## Course Objectives

Lecture Objectives:

1. Describe the principles and advantages of object-oriented programming.
2. Analyze a problem statement and construct appropriate classes to model the problem.
3. Describe the differences between the public, private, protected, and friend access permissions.
4. Design programs that take advantage of the modularity of object-oriented programs.
5. Evaluate and modify programs using C++.
6. Use object-oriented language concepts in software development projects including encapsulation, abstraction, instantiation, overloading, inheritance, composition, and polymorphism.

7. Incorporate file-handling techniques for data input, output, and updating using C++ methods for text and random access file organization.

Laboratory Objectives:

1. Write correct C++ programs that utilize sequencing, conditionals, and loops.
2. Write correct C++ programs consisting of at least two classes.
3. Write correct C++ programs utilizing file I/O to read and write text files.
4. Write correct C++ programs utilizing arrays of primitive types and arrays of objects.
5. Write program documentation conforming to acceptable industry practices.

## General Education Information

- Approved College Associate Degree GE Applicability
- CSU GE Applicability (Recommended-requires CSU approval)
- Cal-GETC Applicability (Recommended - Requires External Approval)
- IGETC Applicability (Recommended-requires CSU/UC approval)

## Articulation Information

- CSU Transferable
- UC Transferable

## Methods of Evaluation

- Objective Examinations
  - Example: Multiple-choice exam question: Which of the following will increment the x variable by 1? a. x++ b. ++x c. x += 1 d. x = x + 1 e. All of the above Answer: e
- Problem Solving Examinations
  - Example: Design and implement a C++ class to represent the main character in a "Flappy Birds" style of video game. The class should have the following methods: void flap(); int getHeight(); boolean isCrashed(); void glide(); Rubric Grading.
- Projects
  - Example: Write a program to keep track of the list of the patients for each doctor at a hospital. implement the following classes: Person, Patient, Doctor, make sure to use the has-1 and is-a relationship. Create a list of the doctors along with the list of the patients for each doctor. Start your project by creating a UML diagram for your classes. In your UML include all the methods that is needed. Rubric Grading.

## Repeatable

No

## Methods of Instruction

- Laboratory
- Lecture/Discussion
- Distance Learning

Lab:

1. Following an instructor discussion on C++ program, students read a written specification for a C++ program. In the hour-long lab session, students write and debug the program which includes conditional statement and classes.. The instructor circulates among the students, offering assistance and Socratically questioning student design choices. (Laboratory Objectives 1 & 2)

Lecture:

1. Students have read the chapter about polymorphism before class. The instructor presents several examples of polymorphic data types. Students and instructor debate the implementation strategies, focusing on design tradeoffs. (Lecture Objective 6)

Distance Learning

1. Students have read the chapter about polymorphism before class. In a video lecture the instructor presents several examples of polymorphic data types. instructor explains the implementation strategies, focusing on design. The instructor posts a quiz on the topic covered. The quiz relates to the reading material and the lecture and helps identify potential trouble areas in student understanding. (Lecture Objective 6)

- Author: Josuttis, Nicolai
- Publisher: Addison Wesley
- Publication Date: 2012
- Text Edition: 2nd
- Classic Textbook?: No
- OER Link:
- OER:
- Murach's C++ programming
  - Author: Joel Murach
  - Publisher: Mike Murach and Associates
  - Publication Date: 2018
  - Text Edition: 1st
  - Classic Textbook?: No
  - OER Link:
  - OER:

## Typical Out of Class Assignments

### Reading Assignments

1. Read the textbook section on "compound data types" to learn about the differences between pointers and references. Be prepared to discuss it during the lecture. 2. Locate and read online documentation for the Clang compiler. From this to determine how to invoke the compiler so that "loop unrolling" optimization is performed. Be prepared to discuss in class.

### Writing, Problem Solving or Performance

1. Using a Car Rental business as a guide, create a class called "Invoice" with 6 private data members: invoice number, days rented, insurance amount, rental rate, total fee, total charge. Create 3 member functions: GetData(), Calcfee(), and ShowInvoice(). Instantiate the object in main() and call the member functions. 2. Create an overloaded minus sign "-" operator to handle a discount coupon. Prompt if there is a coupon, and the percentage discount. Use you overloaded minus operator to recalculate the customer fees.

## Other (Term projects, research papers, portfolios, etc.)

### Required Materials

- The C++ Programming Language
  - Author: Stroustrup, Bjarne
  - Publisher: Addison Wesley
  - Publication Date: 2013
  - Text Edition: 4th
  - Classic Textbook?: No
  - OER Link:
  - OER:
- C++ Primer
  - Author: Lippman, Stanley, et al
  - Publisher: Addison Wesley
  - Publication Date: 2012
  - Text Edition: 5th
  - Classic Textbook?: No
  - OER Link:
  - OER:
- The C++ Standard Library

**Other materials and-or supplies required of students that contribute to the cost of the course.**