

Last Revised: September 2022

Last Reviewed: September 2022

COURSE INFORMATION**Course Title:** C++ for Programmers**Course Number:** CSCI 200**Credits:** 3**Total Weeks:** 14 (Fall, Spring)
12 (Summer)**Total Hours:** 39**Course Level:** First Year Second Year
 New Revised Course
 Replacement Course**Department:** Computer Science **Department Head:** M. O'Connor **Former Course Code(s) and Number(s) (if applicable):** N/A**Pre-requisites (If there are no prerequisites, type NONE):** CSCI 125 or equivalent**Co-requisite Statement (List if applicable or type NONE):** NONE**Precluded Courses:** N/A**COURSE DESCRIPTION**

This course is designed to extend the student's knowledge of the principles and practice of Object-Oriented Programming (OOP) to the C++ programming language. Students must have a thorough understanding of the features of OOP before starting this course. The course begins with a review of basic programming techniques and OOP concepts and progresses to advanced topics in OOP using C++. The examples and exercises require knowledge of fundamental algorithms and programming techniques in an object-oriented context

LEARNING OUTCOMES

Upon successful completion of the course, students will be able to:

- Outline the essential features of the C++ programming language.
- Describe the fundamental principles of Object-Oriented program design.
- Describe the basic concepts of Software Engineering.
- Define algorithms using pseudocode.
- Construct C++ programs from algorithms using standard OOP methodologies.
- Apply the concepts of class, method, constructor, and object.
- Apply the concepts of program data abstraction, function abstraction, inheritance, overriding, overloading and polymorphism.
- Describe and apply techniques to debug C++ programs.
- Trace the execution of C++ programs.
- Use elementary data structures such as arrays and linked lists.
- Implement fundamental algorithms such as the linear search and selection sort.
- Use C++ standard templates.
- Test programs using the methods described in the course.
- Document a project.

INSTRUCTION AND GRADING

Instructional (Contact) Hours:

Type	Duration
Lecture	39
Seminars/Tutorials	
Laboratory	
Field Experience	
Other (<i>specify</i>):	
Total	39

Grading System: Letter Grades Percentage Pass/Fail Satisfactory/Unsatisfactory Other

Specify passing grade: 50%

Evaluation Activities and Weighting (total must equal 100%)

Assignments: % <i>Specify number of, variety, and nature of assignments:</i>	Lab Work: 15%	Participation: % <i>Specify nature of participation:</i>	Project: % <i>Specify nature of project:</i>
Quizzes/Test: 20%	Midterm Exam: 30%	Final Exam: 35%	Other: %

TEXT(S) AND RESOURCE MATERIALS

Provide a full reference for each text and/or resource material and include whether required/not required.

Required: Starting out with C++ From Control Structures through Objects 9th Edition by Tony Gaddis

Reference: Free online book: Thinking in C++ 2nd Edition by Bruce Eckel:

<https://ia802806.us.archive.org/25/items/ThinkingInCVol1/Thinking%20in%20C++%20-%20Vol%201.pdf>

COURSE TOPICS

List topics and sequence covered.

- Week 1 Introduction to Computers and Programming, and C++
- Week 2 Expressions and Interactivity
- Week 3 Conditional Statements in C++: Making Decisions
- Week 4 Loops and Files in C++
- Week 5 Functions in C++

Week 6	Arrays and Vectors in C++
Week 7	Searching and Sorting Arrays in C++ Midterm
Week 8	Pointers, Characters, and Strings
Week 9	Structured Data
Week 10	Classes in C++
Week 11	Classes in C++
Week 12	Inheritance, Polymorphism, and Virtual Functions
Week 13	Exceptions, Templates and Linked Lists
Week 14	Final Exam

NOTES

1. Students are required to follow all College policies. Policies are available on the website at: [Coquitlam College Policies](#)
2. To find out how this course transfers, visit the BC Transfer Guide at: bctransferguide.ca