

Advanced Programming (C++) – Spring 2018

Course Description

This course introduces students to the fundamentals of structured programming and to the procedural and object-oriented aspects of the C++ programming language. Students will create programs to demonstrate the topics of program control, functions, arrays, pointers, and data structures. Microsoft's Visual C++ will be used as the primary development tool; however, other environments may also be utilized. Emphasis will be placed on the creation of platform-independent applications in order to allow students to become familiar with the core features of the C++ language.

Course Requirements

- All program source code will be documented
- All program source code will contain comments listing the author, assignment number/title, and date
- You are required to keep a notebook for this course. Your notebook shall contain the following:
 - All class handouts
 - All lab exercises and source code
 - All quizzes
 - All homework assignments
- Your notebook will have a section for each category of item as described above. All material placed in your notebook will be dated and in order
- All program source code will be kept in a notebook in order

Desired Learning Outcomes

At the completion of this course, students will be able to:

1. Work effectively in the Visual C++ program development environment.
2. Develop algorithms by applying concepts of structured programming.
3. Design classes to solve a variety of problems.
4. Demonstrate the use of fundamental C++ elements including variables, operators, statements, and pointers.
5. Implement the use of standard programming structures including decision, repetition, functions, arrays, and classes.

Class Methodology

The class will be comprised of a combination of lecture, discussion, exercise, reading, and projects. Students are expected to come to class each day fully prepared to participate in the days activities. In this course, you will be apply fundamentals that you learn by developing solutions to a variety of programming challenges.

Textbook

Starting Out with C++, Tony Gaddis
Various E-Books

Dual Enrollment Credit

Credit for CIS158 – Introduction to C++ Programming is offered for this course through Great Bay Community College.

Topic List

<ol style="list-style-type: none">1. Introduction to C++<ol style="list-style-type: none">a. The cout Objectb. The #include Directivec. Variables, Constants, and the Assignment Statementd. Identifierse. Integer Data Typesf. The char Data Typeg. The C++ string Classh. Floating-Point Data Typesi. The bool Data Typej. Size of a Data Typek. Scopel. Arithmetic Operatorsm. Comments
<ol style="list-style-type: none">2. Expressions and I/O<ol style="list-style-type: none">a. The cin Objectb. Mathematical Expressionsc. Implicit Type Conversiond. Explicit Type Conversione. Overflow and Underflowf. Named Constantsg. Multiple and Combined Assignmenth. Formatting Outputi. Working with Characters and String Objectsj. Using C-Stringsk. Working with Files
<ol style="list-style-type: none">3. Decision Making Structures<ol style="list-style-type: none">a. Relational Operatorsb. The if Statementc. The if/else Statementd. Menu-Driven Programse. Nested if Statementsf. Logical Operatorsg. Comparing Characters and Stringsh. The Conditional Operator 206i. The switch Statement

4. Repetition Structures
a. The Increment and Decrement Operators
b. The while Loop
c. Counters
d. The do-while Loop
e. The for Loop
f. Sentinels
g. Using a Loop to Read Data from a File
h. Nested Loops
i. Breaking Out of a Loop
j. The continue Statement 281
5. Functions
a. Modular Programming
b. Defining and Calling Functions
c. Function Prototypes
d. Sending Data into a Function
e. Passing Data by Value
f. The return Statement
g. Returning a Value from a Function
h. Local and Global Variables
i. Static Local Variables
j. Default Arguments
k. Using Reference Variables as Parameters
l. Overloading Functions
m. The exit() Function
6. Classes and Objects
a. Abstract Data Types
b. Object-Oriented Programming
c. Classes
d. Objects
e. Member Functions
f. Constructors
g. Destructors
h. Private Member Functions
i. Passing Objects to Functions
j. Object Composition
7. Arrays
a. Accessing Array Elements
b. Inputting and Displaying Array Contents
c. Array Initialization
d. Processing Array Contents
e. Using Parallel Arrays
f. Arrays as Function Arguments
g. Two-Dimensional Arrays 526
h. Vectors
i. Arrays of Class Objects

8. Searching, Sorting and Algorithm Analysis
a. Search Algorithms
b. Searching an Array of Objects
c. Sorting Algorithms
d. Sorting an Array of Objects
e. Sorting and Searching Vectors
f. Analysis of Algorithms
9. Pointers
a. Pointers and the Address Operator
b. Pointer Variables
c. The Relationship Between Arrays and Pointers
d. Pointer Arithmetic
e. Initializing Pointers
f. Comparing Pointers
g. Pointers as Function Parameters
10. More About Characters, Strings, and the string Class
a. C-Strings
b. Library Functions for Working with C-Strings
c. Conversions Between Numbers and Strings
d. Character Testing
e. Character Case Conversion 797
f. More About the C++ string Class
11. Templates
a. Function Templates
b. Class Templates
c. Standard Template Library (STL)
12. Data structures
a. Linked List
b. Stacks and Queues
c. Binary Trees
13. Computer Architecture
a. Machine Architecture
b. Machine Language
c. Assembly Language
14. Number Representation in Computer Systems
a. Sign-Magnitude
b. IEEE 754

Tentative Schedule

Week	Topics	Reading Assignment
1	Introduction to C++	Chapter 2
2	Expressions and Interactivity	Chapter 3
3	Making Decisions	Chapter 4
4	Looping	Chapter 5
5	Functions	Chapter 6
6	Functions	Chapter 6
7	Introduction to Classes and Objects	Chapter 7
8	Introduction to Classes and Objects	Chapter 7
9	Arrays	Chapter 8
10	Searching, Sorting, Recursion and Algorithm Analysis	Chapter 9 Chapter 14
11	Pointers	Chapter 10
12	More Classes and Object Oriented Programming	Chapter 11
13	Template Classes	Chapter 16
14	Linked Lists	Chapter 17
15	Stacks and Queues	Chapter 18
16	Binary Trees	Chapter 19

Course Grading

Participation	3 Daily Project Grades
Homework	1 Daily Project Grade
Quizzes	1 Daily Project Grade
Projects/Labs	Daily Project Grade
Portfolio	4 Daily Project Grades

Grading Standards

It is expected that you do your best on all project/lab activities. I generally assign one of four possible grades to your labs.

- √+ all project requirements completed in an exemplary manner (100)
- √ most project requirements completed in an acceptable manner (85)
- √- minimal project requirements completed in a substandard manner (70)
- 0 project not completed or completed in an unacceptable manner (0)

Classroom Rules

- Each class period will start at your desk. Do not go to your computer until you are told to do so.
- We have a lot of material to cover in our 90 minutes together. You are to be involved in productive work that entire time. Do not stand by the door waiting for the bell to ring.
- If you need to leave the classroom, you must sign out (legibly). If I can not read your signout, you will lose that privilege.
- No food at computers. If you need a snack, eat it at the desk. If the classroom is not kept clean, there will be no food or drink in the classroom.
- If you make a mess, you are responsible to clean it up.
- No downloading of files; this includes the installation of files from USB drives.
- No games!!!
- Keep a class notebook of notes, handouts, assignments, tests, quizzes, etc.
- Keep your professional portfolio up-to-date
- Come to class prepared. This means you must have a writing instrument and paper top take notes on.
- **No assignments will be accepted more than 5 days after they are assigned.**
- See me for any exceptions to the above rules

Things you need to buy for class:

1. One 1-inch notebook (for Professional Portfolio)
2. One 3-inch notebook (for class notes and handouts)
3. A writing instrument

Recommended:

1. A Flash drive to transport work to and from school

Plagiarism and Collaboration

When working on various projects in this class, you are encouraged to collaborate at some level. Plagiarism is offering someone else's work as your own, whether one sentence, whole paragraphs, or blocks of code. Whether from an internet source, book, periodical, the writing of other students, or source code shared over the network. It is also dishonest to submit your own paper (or program) as original work in more than one course. There is a fine line sometimes between plagiarism and collaboration. Plagiarism is unacceptable here or at any time in your future career. Plagiarism will not be tolerated. In this class, plagiarized work will receive a grade of 0. **It is unacceptable for one person to write a program and share it over the network with other students placing your name on code that you did not write.** If you have any question as to whether or not you are plagiarizing someone's work, ask!!! Better ask for clarification than receive a 0 on your test.

Things you need to do:

- Install Visual Studio (with C++) on your home computer