

CET U301
Introduction to C++ Programming (4 SH)**Summer 1 2005***Instructor:*Randal August, Assistant Professor, raugust@coe.neu.edu, 263 SN, 617-373-2064*Textbook Requirements:**Problem Solving with C++, The Object of Programming*, 5th edition, Walter Savitch, Addison-Wesley, ISBN: 0-32-128833-5*Prerequisites:*

CET U201 / CET E201 or Permission from faculty advisor

Co-requisites:

CET U302

Course Description:

This first course in C++ programming will introduce student to the language, its history, its applications and its implementation. Rudiments of the language are presented along with a detailed explanation of how different programming constructs are used. Weekly programming assignments complement laboratory exercises allowing students to learn about primitive data types, arrays, structures, program control details, strings, pointers and C++ classes. (*Quarter Equivalent:* GET 1100 / GET 4100)

Course Goals:

To provide a foundation of C++ syntax and grammar to include declaration and use of class objects.

Basis for Course Grade:

The course consists of lectures on Friday and laboratory on Tuesday. Attendance at all classes and laboratories is mandatory and counts as part of the class participation portion of the grade. Students absent for legitimate reasons are required to seek alternative sources to retrieve missed notes. Homework assignments will be assigned during Lab class and are due by 6:00 PM on the following Tuesday.

| Content | Total |
|---|--------------|
| Class Participation | 5% |
| 10 Homeworks (2 at 1%, 1 at 3%, 6 at 5%, 1 at 10% each) | 45% |
| 3 Quizzes (10% each) | 30% |
| Final | 20% |
| Total: | 100% |

University's academic honesty policy: Northeastern University is committed to the principles of intellectual honesty and integrity. All members of the Northeastern community are expected to maintain complete honesty in all academic work, presenting only that which is their own work in tests and assignments. If you have any questions regarding proper attribution of the work of others, contact your professor prior to submitting the work for evaluation.

Course Objectives:

- Demonstrate familiarity with the basics of the language, the syntax, grammar and semantics of C++ programs through the use of a simple “boilerplate” program.
- Demonstrate an ability to use an integrated development environment / tool and describe the basic functions of each of the components.
- Understand the fundamental characteristics and functions of the control structures used perform operations in structured program.
- Understand the use of Boolean expressions and the logical operations of AND, OR, NAND, NOR, XOR and NEGATION in decision making and program control.
- Identify and describe the elements of a basic program including preprocessor directives, library / include files, functions / methods parameters, data and variable types that make up a structured program.
- Given a defined problem involving data entry develop an algorithm to solve the problem, organize the data using structures with multiple data types and produce an output.
- Demonstrate the ability to debug and test a program of student design, using the integrated development environment / tool.
- Given a simple, but “bug – ridden” program, utilize the capabilities of the integrated development environment in order to analyze and correct the errors to produce the expected output.
- Using a verbal description of a problem develop an object class organization of the data and requisite methods to manipulate the data contained within an instance of the data.
- Given a program of moderate difficulty, modify the solution so that class declarations and algorithm are separated into independent and reusable modules. Enhance the class declaration(s) through the use of inheritance without altering the base class declaration.
- Create the necessary modifications to a class to allow for the overload of common operator symbols such as + (addition) and - (subtraction).
- Given a problem statement of moderate difficulty, develop a program that will perform file I/O and utilize a C++ class object designed specifically for the problem solution using arrays to store a large amount of data within the program.

General Course Schedule:

| Topics | Readings | Topic description / Homework Assignments |
|---------------|------------------------------------|---|
| 1 | Chapter 1 | Basic relationship between hardware and software, the software design process, and some history of the language. Make a quick read of all sections. |
| 2 | Chapter 2 section 2.1 to 2.3 | Language basics, variable declarations and simple input and output operations |
| 3 | section 2.4 and 2.5 | Program flow control and decision making, program style and readability. |
| 4 | Chapter 3 section 3.1 to 3.3 | Top down design, predefined and user defined functions / methods. |
| 5 | section 3.4 and 3.5 | Procedural abstraction, local and global variables. |
| 6 | section 3.6 | Overloading function names and conversion of data types. |
| 7 | Chapter 4 section 4.2 and 4.4 | Function design and use of parameters. Testing and debugging functions and methods. |
| 8 | section 4.1 and 4.3 | The void data type, procedure design and abstraction. |
| 9 | Chapter 5 section 5.1 to 5.3 | File operations, redirection of I/O, data formatting. |
| 10 | Chapter 6 section 6.1 | C Structures, the organization of data. |
| 11 | section 6.2 and 6.3 | C++ Classes, the connection of data and methods. |
| 12 | Chapter 7 section 7.1 and 7.2 | Flow control and multi-way branches with complex Boolean expressions. |
| 13 | section 7.3 and 7.4 | Design and implementation of repeating operations. |
| 14 | Chapter 8 section 8.2 | Function overloading |
| 15 | Chapter 10 section 10.1 to 10.4 | Arrays, storage of large amounts of data |

