

**Concordia University
Department of Computer Science
and Software Engineering**

**Advanced Program Design with C++
COMP 345 --- Fall 2013**

Contact Information

Professor in charge : Joey Paquet, paquet@encs.concordia.ca
Web page : newton.cs.concordia.ca/~paquet/wiki/index.php/COMP345_fall_2013
Lectures : Section D -T-J--- 13:15 - 14:30 FG B070 Paquet, J. paquet@encs.concordia.ca
Section N M-W---- 16:15 - 17:30 H 629 Taleb, M. mtaleb@encs.concordia.ca

Calendar Description

Prerequisite: COMP 352 previously or concurrently. Introduction to C++. I/O with stream classes. Pointers and their uses. The Standard Template Library (STL): containers, algorithms, iterators, adaptors, function objects. Class design: constructors, destructors, operator overloading, inheritance, virtual functions, exception handling, memory management. Advanced topics: libraries, locales, STL conventions, concurrency, template meta-programming. Applications of C++: systems, engineering, games programming. Project. Lectures: three hours per week. NOTE: Students who have received credit for COMP 446 may not take this course for credit.

Rationale

Most of our courses are taught using the Java programming language. C++ programming is pervasive in many key areas of the software industry. Though C++ and Java have many similar syntactical elements and structures, C++ has many subtleties and features that differ from Java. This course aims at teaching C++ to an audience well-trained in computer programming and putting the newly acquired knowledge into practice through a challenging project.

Prerequisite knowledge

It is assumed that all students have extensive experience with computer programming, though no prior knowledge or experience of C++ is assumed.

Project

The project is to be undertaken small teams of exactly 4 members and consists of the building of a challengingly large C++ program. The completion of the project is divided into two separate components: (1) the *Intermediate Project Delivery* is a first operational build of the software, effectively demonstrating the full implementation of some important software features; (2) the *Final Project Delivery* is the demonstration of the finalized version of your software. During the final project delivery, you also have to demonstrate that your code includes many of the C++ features presented in the lectures. The individual assignments will also be related to the project.

Evaluation

| | |
|--------------------------------|-----------------|
| Individual: | |
| Examinations (midterm & final) | 15% + 30% = 45% |
| Assignments (3) | 3 X 5% = 15% |
| Team: | |
| Intermediate Project Delivery | 15% |
| Final Project Delivery | 25% |

Textbook (non mandatory)

Walter Savitch. *Absolute C++*. 5th edition, Addison-Wesley, 2013.

Graduate attributes

As part of the Software Engineering program curriculum, the content of this course includes material and exercises related to the teaching and evaluation of *graduate attributes*. Graduate attributes are skills that have been identified by the Canadian Engineering Accreditation Board (CEAB) as being central to the formation of Engineers. This particular course aims at teaching and evaluating 3 of the 12 CEAB graduate attributes. The following is a description of these attributes, along with a description of how these attributes will be incorporated in the course.

(1) **Problem analysis** is the ability to use appropriate knowledge and skills to identify, analyze, and solve complex engineering problems in order to reach substantiated conclusions. The project in this course is defined in such a way that requires the students to analyze the problem at hand before and determine for themselves exactly what needs to be done, and then determine how and with the help of what tools and software libraries it can be achieved.

(2) **Design** is the ability to design solutions for complex, open-ended engineering problems and to design systems, components or processes that meet specified needs with appropriate attention to health and safety risks, applicable standards, and economic, environmental, cultural and societal considerations. The project in this course is presented in an open-ended fashion, and its size and complexity is such that it needs to be tackled in teams of 3. The individual assignments provide a platform for designing at a smaller level, and provide the additional difficulty of having to be integrated in the larger design of the project.

(3) **Use of Engineering tools** is the ability to create, select, apply, adapt, and extend appropriate techniques, resources, and modern engineering tools to a range of engineering activities, from simple to complex, with an understanding of the associated limitations. The course teaches the use of the C++ language, and leaves the students free to select what programming environment and libraries that they will use in the assignments and project. Selection and use of the right tools and libraries is a crucial aspect of accomplishing the practical work.