Concordia University
Department of Computer Science
and Software Engineering

Advanced Program Design with C++
COMP 345 --- Fall 2015

Contact Information

Professor in charge :   Joey Paquet, paquet@encs.concordia.ca
Web page :   www.cse.concordia.ca/~paquet/
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   FG B070   Sadri, J. j_sadri@encs.concordia.ca

Calendar Description

COMP 345 - Advanced Program Design with C++ (4 credits) Prerequisite: COMP 228 or SOEN 228; COMP 352. 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 metaprogramming. Applications of C++: systems, engineering, games programming. Project. Lectures: three hours per week. Laboratory: two hours per week.

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%

Textbooks (non mandatory)

Y. Daniel Liang, Introduction to Programming with C++. Third Edition, Prentice-Hall, 2014. ISBN-13: 978-0-13-325281-1
Walter Savitch, *Absolute C++*. Fifth Edition, Addison-Wesley, 2013. ISBN-13: 978-0-13-283071-3
Walter Savitch,Problem Solving with C++. Ninth Edition, Pearson, 2014. ISBN-13: 978-0-13-379174-3
Bjarne Stroustrup, A Tour of C++. Addison-Wesley, 2014. ISBN-13: 978-0-321-958310
Bjarne Stroustrup, The C++ Programming Language. Fourth edition. Addison-Wesley, 2013. ISBN-13: 978-0-321-56384-2

Graduate attributes

As part of both the Computer Science and 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) and the Canadian Information Processing Society (CIPS) as being central to the formation of Engineers, computer scientists and information technology professionals. As such, the accreditation criteria for the Software Engineering and Computer Science programs dictate that graduate attributes are taught and evaluated as part of the courses. This particular course aims at teaching and evaluating 3 graduate attributes. The following is a description of these attributes, along with a description of how these attributes will be incorporated in the course.

Knowledge-base: Application of advanced programming principles using a mid-level programming language. Use of design patterns and architectural design. Manual/explicit memory management. Language/compiler/runtime system implementation concepts and details.

Design: 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 4. 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.

Use of tools: 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.

Individual and team work: Work in teams of 3-5 members on a large project, as well as on individual assignments also related to the project.

Communication skills: Proper code documentation using code comments, as well as simple description of design decisions. Oral presentations for project deliveries.