COMP322: Introduction to C++ - Winter 2012
Monday 14:35-15:25, Trottier 1100

1   General Information

This one-credit course will provide a very basic introduction to the C++ programming language.

The official course URL is `http://www.cs.mcgill.ca/~dpomer/comp322/winter2012/`. This page will be updated with announcements, lecture notes and assignments.

The instructor for this course is Dan Pomerantz. I can be reached most efficiently by email at `dpomer@cs.mcgill.ca`. My office is McConnell 306. I will do my best to respond to emails within 24 hours.

While I may reschedule as the semester progresses, initially I will have office hours on Tuesdays from 10:30-11:30. I'll also be available after class in most cases and by appointment.

The teaching assistant information will be announced on the course webpage when it is finalized.

2   Description

C++ is a popular object-oriented programming language, originally developed at Bell Labs by Bjarne Stroustrup from 1979-1983. C++ is descended from the C programming language (Kernighan and Ritchie, 1978) and is an ancestor of the Java language (Gosling, Joy and Steele 1996). This course will cover the essential features of the C++ programming language. We will focus on the features that distinguish C++ from Java (pointers, memory allocation, templates) and C (classes, operator overloading, namespaces, exceptions). We will touch on the some of the services provided by the C++ standard library. Given the limited time available in lecture, we cannot attempt to cover every single concept of C++.

3   Prerequisites

This course requires that you have already completed at least one of COMP202, COMP206, COMP208, or COMP250. You should be very familiar with the C programming language. If you are familiar with Java but not C, you should be OK but will find you are behind on a couple concepts (C-style pointers) and ahead on others (object oriented programming). As C++builds on C (hence the name C++) we'll often look at examples comparing and contrasting C++with C. That said, Java is very similar to C syntactically, so knowledge of Java will get you along as well.

Some key differences between Java and C:

1. C does not have any notion of *objects*. (C++of course, does). The standard library of C is not nearly as rich as Java.

2. Java does much more to abstract the notion of pointers/references than C does.

3. C allows you to do many things that Java simply does not. For example, in Java, the following statement would cause a compiler error. `int x = 3.5;`. In C, no such error occurs, rather

the value is just truncated. In many other cases, Java programs have a fixed response whereas in C they may work sometimes but not others. For example, in Java an array out of bounds error causes the program to crash. In C, it will sometimes crash and sometimes continue to proceed as normal.

This means to program in C you have to be extra careful to avoid mistakes as they are more difficult to find.

4. There are several minor syntactical differences between the two.

If you have not met all of these prerequisites, you should not take this course. If you are concerned about this or worried about your knowledge of C, please discuss with your instructor.

While we do not have time to turn this into a full object-oriented (OO) programming course, I will attempt to explain OO principles at least as they apply to C++.

4 Goals of the course

At the end of this course, you should:

1. Understand the basics of object oriented programming. (Those who have worked with Java before may already understand this)

2. Know some of the many tricks of C++and further to have an idea of *when* to apply these tricks and when to stick with the basics

5 Course Resources

There will be two primary locations for course resources: WebCT and the course webpage. It is your responsibility to check both of them in order to get up to date information on assignments, quizzes, lectures, announcements, etc. The course webpage can be found at `http://www.cs.mcgill.ca/~dpomer/comp322/winter2012/index.html`

All assignments will be submitted on WebCT. In addition, there will be discussion boards on WebCT which will be monitored by the instructor and TAs. You are encouraged to post questions here as then other students can help as well and share their knowledge. You may not post your code on WebCT, but you may post small snippets of code (e.g. 10 or so lines) in order to clarify why something isn't working. When in doubt, err on the side of too few lines of code and perhaps your instructor will request you post more to clarify.

6 Textbook

There is no formal textbook for the class. However, I recommend that you obtain one of the standard reference texts on C++. One such book is *The C++ Programming Language: 3rd Edition* by Bjarne Stroustrup.

Another highly recommended book is *Accelerated C++* by Andrew Koenig and Barbara Moo. You can purchase this at `http://www.amazon.ca/Accelerated-C-Practical-Programming-Example/dp/020170353X`.

7 Evaluation

There will be four short homework assignments and two quizzes.

You will have roughly two or three weeks to complete each assignment (this may vary slightly). As the purpose of the class is to familiarize you with the C++language, assignments will consist mostly of programming problems. An important component of any programming work is documentation, therefore you should use comments throughout the code to explain the reasoning behind your design choices. Both the quality and functionality of the code and the completeness and accuracy of your comments will be taken into account in the grading of assignments.

You are welcome to discuss assignments with your fellow students, but all submissions must reflect individual work.

Assignments *must* be submitted electronically via WebCT, by the due date. Late assignments will be penalized by 10 percent per day (i.e. an assignment received 25 hours late will be eligible for at most 80% of the possible score).

The quizzes will be given in class, and will consist of roughly 10 short answer or multiple-choice questions. On quiz days, assume you will have about half the class time to complete the quiz.

Each assignment or quiz will be worth 20% of the total grade. The lowest assignment or quiz score will be dropped from this calculation. The other five assignment or quiz scores will be used to calculate a final score out of 100%. If you miss a quiz for an excused absence, you have the option of dropping the missed score. If you choose to drop the score, your final grade would be calculated using the four assignments and one quiz. Special arrangements will be made for students who must miss both quizzes for legitimate reasons.

Note that if the score on an assignment or quiz is reduced because the work submitted violates McGill's academic integrity guidelines, that score will *not* be dropped. For example, if an assignment is copied from another student or the Internet, the student will receive a score of zero and that score will be included in the calculation of the final grade.

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

8 Approximate Schedule

Note that this is a *tentative* schedule only. It may be revised as the semester progresses. Note that the dates for the assignments in particular may change and won't necessarily be due on the Monday.

1. 09 Jan - Course introduction, some basics of C++

2. 16 Jan - Basic language features (Assignment 1 out)

3. 23 Jan - Pointers and references

4. 30 Jan - Memory management

5. 06 Feb - Input/output using the Standard Library (Assignment 1 due, Assignment 2 out)

6. 13 Feb - Quiz 1, Standard library basics

7. 20 Feb - **Study break**

8. 27 Feb - Classes (Assignment 2 due, Assignment 3 out)

9. 05 Mar - Overloading, Inheritance

10. 12 Mar - Inheritance continued

11. 19 Mar - Exceptions (Assignment 3 due, Assignment 4 out)

12. 26 Mar - Templates

13. 02 Apr - Quiz 2, Templates continued

14. 09 Apr - More on the Standard Library

15. 16 Apr - Finish the Standard Library (Assignment 4 due)

9 Academic Integrity

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples renseignements, veuillez consulter le site www.mcgill.ca/integrity).