

# COMP 322: Intro to C++ (Winter 2015)

## Course Outline

This one-credit course is intended to provide a very basic introduction to the C++ programming language, albeit ambitious students will be able to get more out of the experience. C++ is a popular object-oriented programming language, originally developed at Bell Labs by Bjarne Stroustrup, from 1979-1983, in an effort to combine C's efficiency and flexibility for systems programming and Simula's program organization facilities. That is, C++ attempts to stay close to the hardware in terms of services, and close to the programmer in terms of software design. C++ is also an ancestor of the Java language, which goes away from the hardware in the interest of more comprehensive and secure software. We will focus on features that distinguish C++ from Java (pointers, memory allocation, templates) and C (classes, operator overloading, namespaces, exceptions). We will touch on 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++.

## 1 General Information

The official course URL is <http://www.cs.mcgill.ca/~gcoman/comp322/winter2015/>. The page will be updated with announcements, lecture notes and assignments.

The instructor for this course is Gheorghe Comanici. It is best to reach me by e-mail at [gcoman@cs.mcgill.ca](mailto:gcoman@cs.mcgill.ca). My Office is McConnell 111, and my office hours are Wednesday 10-11:30AM. I will also be available after class and by appointment. Information about the TAs for this course will be posted online.

Please do come and see both the instructor and the teaching assistants during office hours. Do not feel that you are infringing on our time - office hours represent time has been reserved specifically to see you. If you would like to discuss a program, please bring a copy of the code or a laptop with the code on it. We cannot figure out what is wrong with a program just from a description of its behaviour.

## 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 of concepts( C-style pointers). As C++ builds on C (hence the name) we'll often look at examples comparing and

contrasting C++ with C. That said, Java is syntactically very similar, so knowledge of Java will get you along as well. Here are some key differences between Java and C:

1. C does not have any notion of *objects* (C++ 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.
3. C allows you to do many things that Java simply does not. For example, `int x = 3.5;` does not compile in Java, but in C the value simply gets truncated. In many other cases, Java programs have a fixed response, while in C they may work some times, but not others. Take the example of an “array out of bounds error”: Java will always crash, but in C it will sometimes crash and other times continue to proceed as normal. *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 the prerequisites, you should not take the course. If you are worried about your knowledge of C, please discuss with your instructor.

## 2 Goals of the course

At the end of the course, you should:

- Understand the basics of object oriented programming. (Those who have worked with Java before may already understand this)
- Know some of the many tricks of C++ and have an idea of *when* to apply these tricks and *when* to stick with the basics.

## 3 Resources

There will be two primary locations for course resources: myCourses 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.

All assignments should be submitted on myCourses. You are encouraged to post questions on the assignments on the Discussion Board of myCourses. The instructor and the TAs will monitored these discussions, but all students are encouraged to help and share there knowledge. You **may not** post your code on myCourses. If a small snippet of code is really necessary to clarify why your code is not working, post a few lines of code and the instructor might ask you to provide more.

## 4 Textbook

There is no formal textbook for this class, but there are many online resources for learning C++ (check the course website for specific examples), as well as a few standard reference texts on C++. Here are a few that we recommend:

- *The C++ Programming Language: 3rd Edition* by Bjarne Stroustrup.
- *Accelerated C++* by Andrew Koenig and Barbara Moo.

## 5 Evaluation

There will be four short homework assignments and two quizzes. Your top three homework assignments will be each worth 20% of your grade (your worst assignment will **not** be considered) and each quiz will be worth 20%. As the only way to learn a programming language is by practicing it, all homework assignments consist of programming problems. **We expect that you comment your programs thoroughly.** When grading your code we will equally consider the quality/functionality of your code as well as the completeness/accuracy of your comments.

You are encouraged to discuss assignments with your fellow students and we believe collaboration is an important factor in learning and practicing a new programming language. As such, each assignment is organized as follows:

1. 0-credit question - we will not grade it, but we will provide feedback (if you comment it properly and ask for it). You can get code for this question from any resource of your choice (your head, imaginary friend, colleague, Stack Exchange, etc.). *Please make sure you understand the solution, yours or not, as it will be essential for the second part of the assignment.*
2. A few other questions which we will grade. The code submitted for these questions *must reflect individual work* (see **Academic Integrity** section below for more details). Note that you can only get full marks for these questions if the 0-credit code is functional itself. Points will be taken off even if your code is correct but not functional due to an error in the 0-credit question.

We will only grade assignments that satisfy all three conditions below:

- submitted via myCourses (no e-mail submissions)
- submitted before the deadline (submissions which are up to 24 hours late will incur a 30% penalty; every other late submission will be ignored)
- code compiles with a set of header files that we provide (method signatures must be consistent; you will learn about header files in the first lecture)

**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.**

## 6 Tentative schedule

Lecture 1	Jan 08	Introduction and basics	
Lecture 2	Jan 15	Basic Language Features	HW1 out
Lecture 3	Jan 22	Pointers and references	
Lecture 4	Jan 29	Memory management	
Lecture 5	Feb 05	Input/Output	<b>HW1 due ; HW2 out</b>
Lecture 6	Feb 12	<b>Quiz 1</b>	
Lecture 7	Feb 19	Classes	
Lecture 8	Feb 26	Overloading and Inheritance	<b>HW2 due ; HW3 out</b>
	Mar 05	<b>Study break</b>	
Lecture 9	Mar 12	Inheritance (cont.)	
Lecture 10	Mar 19	Exceptions	<b>HW3 due ; HW4 out</b>
Lecture 11	Mar 26	Templates	
Lecture 12	Apr 02	Standard Library	
Lecture 13	Apr 09	<b>Quiz 2</b>	<b>HW4 due</b>

## 7 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.cs.mcgill.ca/integrity](http://www.cs.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.cs.mcgill.ca/integrity](http://www.cs.mcgill.ca/integrity)).