McGill University

COMP-202A Introduction to Computing I

Fall 2005

webct.mcgill.ca

Time and place

- Days and Times:

Section 1:	Mon., Wed. and Fri.	11:35 AM - 12:25 PM
Section 2:	Tue. and Thu.	10:05 AM - 11:25 AM
Section 3:	Mon., Wed. and Fri.	12:35 PM - 1:25 PM

- Rooms:
  - Section 1: Trottier Engineering Building, Room 1100 (ENGTR 1100)
  - Section 2: Duff Medical Building, Theatre 1 (DUFF THTR 1)
  - Section 3: Trottier Engineering Building, Room 0100 (ENGTR 0100)

- Instructors:
  - Section 1: Nomair Naeem
  - Section 2: Yannick Daoudi (course coordinator)
  - Section 3: Marc Lanctot

Introduction

The School of Computer Science would like to welcome you to the COMP-202 course. The course will be taught using the Java programming language. The purpose of this document is to provide you with an overview of what lies ahead in this course. We shall begin with a brief introduction of the course contents followed by some important general information about the course. Please read this document carefully and keep it for reference throughout the term.

Course Description

This course is intended for those students with little or no background in programming. The main thrust of the course is directed toward learning the fundamental tools in designing and implementing computer programs. In addition, some time will be devoted to learning about the structure of personal computers, networks, and applications of computers in general. The programming problems will be drawn from both scientific and non-scientific applications. All programs must be written in Java.

Students will learn to develop programs using the Java Development Kit (JDK) and a programming development environment, such as JCreator LE or Eclipse. These development environments support all the tools necessary for the rapid development of

Java programs, they lie on top of JDK, making it easier for the students to write, debug and run their Java programs. Although this course is intended to teach students to program in the Java language, the material learned in the course is applicable to programming in other high level programming languages such as C, C++, and Ada. This course will cover both imperative as well as object-oriented programming techniques.

What this course is not about

This course is not about how to use a computer. Here we will not teach you how to send e-mail, how to browse the Web, how to set-up and configure a computer, how to use specific software applications, how to design web pages, or how to deal with operating systems problems. BUT, the course does provide some introductory labs that supply some of this kind of help.

Course Prerequisites and Texts

- Course prerequisite:
    - A CEGEP level math course or equivalent. For students who did not attend CEGEP, any upper-level mathematics course is sufficient.

- Course Text:
    - Java Software Solutions Foundations of Program Design, 4th Edition, Addison-Wesley, by John Lewis, William Loftus. (*Note: you may obtain a copy of this textbook from the McGill bookstore. Copies obtained from the bookstore are packaged with a CD*)

- Other References:
    - Computing Concepts with Java Essentials by Horstmann.
    - The Java Programming Language, Second Edition by Arnold and Gosling.

Course Grading System and Deadline Policy

- Assignments: 30%
- Midterm: 20% (2 hours, Wednesday March 8th, 18:00 – 20:00)
- Final: 50% (3 hours, TBA)

There will be six assignments, each with equal weight. We will calculate your assignment mark by taking your best 5 of 6 assignments, each counting for 6% of your final mark. We encourage you to complete all of the assignments, as this is the major way in which you learn the material. If, for some reason you cannot complete one assignment, then we will assign a 0 to that assignment, and we will count the other 5 completed assignments.

Late assignments will be deducted 5% each day for which they are late. Assignments handed in more than 2 days (including weekend days) after the deadline will not be accepted, nor marked. Assignment #3 and assignment #6 cannot be late at all because we will post the solution in time for the midterm and final. No exceptions will be made.

Scheduled Labs

Lab #	Title	Contents
1.	How to use the lab computers	- Get a password, how to login and out - Basic Unix command - How to save data on a diskette - How to use the Internet and webCT
2.	The development environment	- The IDE, inputting a program - Running & debugging a program - Help with assignment #1
3.	Thinking like a programmer	- Organizing your thoughts, top-down, steps - Solving problems
4.	Programming with Objects	- Using encapsulation - Object-oriented approach to solving problems
5.	Midterm Exam Tutorial	
6.	Final Exam Tutorial	

Information about Labs on Campus

The information in this section is to be used as a general guideline only. We suggest that students contact the work area of their choice, to enquire about the hours, and for any further information needed. Most facilities are available to all McGill students but there are locations with restricted usage permitting access only to those students within the faculty or department indicated. For example, students in the Faculty of Science may want to use the new lab to be set up in Burnside Hall, and Engineering Students may want to use the new Engineering Labs.

A usage fee is required by some facilities and it may be a little more for students not from the faculty or department indicated. A student may pay the fee on a per day basis or become a member, i.e., obtain a pass by paying for a full semester or academic year. Becoming a member may have certain advantages such as, for instance, computer reservation privileges. Most areas provide printing, but may have a charge per page.

Below, we list the information about the Computer Science Lab that can be used by COMP-202 students, for their course work only. Information about other labs can be obtained via the Computing Center's web pages. Please visit the labs for time changes!

Refer to http://www.cs.mcgill.ca/socsinfo/labs/ for more information.

NOTE: Using lab ENGTR 3120, login as `newuser` and password as `newuser`, and then fill out the web form to get your password to our systems. You must be registered for the course.

Usually, the labs are open from 10:00 am to midnight on weekdays, from 12:00 to 8:00 pm on weekends. However, lab ENGTR 3120 or anywhere on the third floor (the windows machines where 202 students normally do their assignments) is open 24 hours a day, 7 days a week, but a consultant will be on duty during the above mentioned times only. The students will have to have their student card specially encoded to access the lab area after 5:00 pm on weekdays as well as on weekends.

It is possible that the schedule will be different during the first week of lectures.

To sum up: if you don't care about a consultant being on duty and if the students have their card properly encoded, they can access the windows lab at any time.

Personal Computers and Lab Software

All programming assignments will be built using the JDK Java compiler on personal computers. You may use the JDK tools directly by editing the files using your favorite text editor. However, if you prefer to use a program development environment, you can use the JCreator LE environment (or any other environment, such as Eclipse).

If you want to work on your own computers, you can download and install both the JDK and JCreator LE (or Eclipse) free of charge:

- JDK:
  - You will be able to download a copy of JDK from http://java.sun.com/j2se/1.5.0/download.jsp (select *JDK 5.0 Update 6*). JDK should be installed first.

- JCreator LE: (available for Windows)
  - You may also get a personal copy of JCreator from http://www.jcreator.com/download.htm (select *JCreator LE version*). During Lab #1, the TA will demonstrate how to use this programming environment. Jcreator should be installed after JDK has been installed, as this will avoid several configuration headaches.

- Eclipse: (available for most operating systems)
  - Eclipse is installed on all our lab computers in the Trottier building, room 3120. During Lab #1, the TA will demonstrate how to use this programming environment. If you would rather use Eclipse over Jcreator, or if you are using another operating system (Mac OS, Linux), you can download it from http://www.eclipse.org/downloads/ (select *Other downloads for 3.1.1* for operating systems other than Windows).

Teaching Assistants (TAs)

All TA's will be available for 1 hour per week, in the ENGTR 3107, to help you with your assignments, and answer questions about the course material. You may ask questions to any TA, not just the one who marks your assignments. TA lab times (office hours) are posted on webCT.

How to submit each assignment

Each assignment will contain directions on how and what to submit. Predominantly this course will be using WEB CT for receiving, submitting and grading assignments. We will discuss in class how to use it. WEB CT will also have a discussion board that will be used to ask questions about assignments. Students, instructors and TAs will scan and post answers to questions. **Please no assignment answers and no code!**

Posting of Course Marks

The course marks will be posted on the WEB CT. The marks will be updated after each assignment and midterm. It is **your responsibility** to check that the marks are correct and to notify your instructor of any errors or missing marks **within one week of getting your grade**.

What comes after this course?

Students who enjoyed this course may want to continue with further studies in Computer Science. If you did well in this course (i.e. you got at least a B+, preferably an A), then you may want to continue as a Computer Science Major. Your next course will be COMP-250, Introduction to Computer Science. If you don't want to do a major, but you are still interested in Computer Science, you can continue with a minor in Computer Science. For a minor program, the next course to take is COMP-203, Introduction to Computing II. In planning your academic program, you should always consult with the appropriate advisor.

Course Coordinator & Instructors

Name:	**Nomair Naeem** – teaching section 1
Office hours:	TBA *(or by appointment)*
Office:	McConnell 234
E-Mail address:	nomair.naeem@mail.mcgill.ca *(for assignment questions, use webCT)*
Webpage:	http://www.cs.mcgill.ca/~nnaeem
Telephone:	398-7071 ext.00076# *(for emergencies only, e-mail is preferred)*

Name:	**Yannick Daoudi** – teaching section 2 & course coordinator
Office hours:	Tuesdays & Thursdays 11:30-13:00 *(or by appointment)*
Office:	McConnell 322
E-Mail address:	ydaoud@cs.mcgill.ca *(for assignment questions, post on webCT)*
Webpage:	http://www.cs.mcgill.ca/~ydaoud/comp202
Telephone:	398-7071 ext.00116# *(for emergencies only, e-mail is preferred)*

Name:	**Marc Lanctot** – teaching section 3
Office hours:	Tuesdays 17:00-18:30 & Fridays 13:30-15:00 *(or by appointment)*
Office:	McConnell 234
E-Mail address:	marc.lanctot@mail.mcgill.ca *(for assignment questions, use webCT)*
Webpage:	http://www.cs.mcgill.ca/~mlanct2
Telephone:	398-7071 ext.00076# *(for emergencies only, e-mail is preferred)*

Emailing Rules: Post all your questions about assignments on the WEB CT message boards for all to see both the questions and answers. You are free to answer each other's questions as well. You are only not permitted to provide solution code – exception: one or two lines of code are okay. You can email the TA or Instructor directly for private matters of course.

Important dates

- Classes Start: Tuesday January 3rd
- Classes End: Monday April 10th
- Final Exam Period: Tuesday April 10th – Friday April 28th
- Spring break: Monday February 20th – Friday February 24th

Policy on Assignments and Plagiarism

Official Policy:

- 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).
- 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).

Plagiarism, under any form, will not be tolerated. Assignments must be done individually; you may not work in groups. If you need help with your assignments, please visit a TA during their lab hours or the instructors during their office hours. Do not rely on friends or tutors to do your work for you. You may not copy another person's work in any manner (electronically or otherwise). Furthermore, you must not give a copy of your work to another person. We will be randomly checking for similarities between programs, and you may be asked to present and explain your program to an instructor or TA.

You must include your name, student number and email address at the top of each program or module that you have implemented. By doing so you are certifying that the work is entirely your own, and you are accepting responsibility for any bugs in the program.

Students who require assistance with their assignments should see the teaching assistants during their lab hours, or students should consult with their instructor during his/her office hours. If, for some reason beyond your control, you are too busy to complete one assignment by the deadline, just take a 0 for that assignment (see marking policy above). If you have partially finished an assignment, you must document what works and what does not work in your program, and hand it in. Students who put their name on programs or modules that are not entirely their own work will receive a mark of 0 for that assignment, and this mark will be counted as one of the 5 assignments marks included in the final grade. In addition, the students involved may be referred to the appropriate Associate Dean who will assess the need for further disciplinary action.

Course Calendar

Week #	Material	Readings (4th ed.)	Events
1	- Components of a computer - Storing and manipulating data - Computer networks	1.1-1.3	Placement quiz
2	- Programming languages & Java - Syntax, semantics & Errors - Variables and assignments - Primitive Data Types - Arithmetic operators & conversions	1.4-1.5 2.1-2.5	Lab 1 Assignment #1
3	- Introduction to objects - The String class - Creating objects - Class libraries - Wrapper classes - Input/Output	2.6 3.1-3.8 *Excluding 3.7*	Lab 2
4	- Flow of control through a method - Boolean expressions - Decision-making statements (if and switch) - Repetition statements (while, do and for)	5.1-5.8	Assignment #2
5	- Objects & classes - Encapsulation & visibility modifiers - Scope - Creating new objects - Parameter passing	4.1-4.4	Lab 3
6	- Program development - Object relationships: aggregation - Static members - Method decomposition and overloading	6.1-6.8 *Excluding 6.5-6.6*	Assignment #3 *(due week 9)*
7	- Object references and aliases - Passing objects as parameters - Interfaces	6.5	Lab 4 Assignment #4 *(due week 11)*
8	**Spring break**		Chill out
9	- Array declaration and use - Multidimensional arrays - The ArrayList class - Command-line arguments	7.1-7.7	Lab 5: Tutorial
10	- Inheritance	8.1-8.5	Midterm
11	- Polymorphism	9.1-9.3,9.6	Assignment #5
12	- Exceptions & file I/O	10.1-10.6	
13	- Recursion	11.1-11.3	Assignment #6
14	- Review & catch-up		Lab 6: Tutorial