

# University of Toronto

## APS105: Computer Fundamentals — Fall 2007

Wael Aboelsaadat, Tom Fairgrieve, W. James MacLean

### General Information

This course is designed to give you an introduction to computer science with emphasis on the development of tools for problem solving in science and engineering, and to provide you with the basics for studies in greater depth in subsequent courses. The course consists primarily of lectures and laboratories. Major topics to be covered in the lectures include: programming in C++, representation of information in a computer, development of good programming techniques, program organization, algorithms, and data structures. The laboratories will provide you with an opportunity to exercise concepts introduced in the lectures through a number of programming assignments using the C++ programming language.

### Instructors

Name	E-Mail & Web	Office
Wael Aboelsaadat	wael@cs.utoronto.ca <a href="http://www.cs.utoronto.ca/~wael/">http://www.cs.utoronto.ca/~wael/</a>	BA4261
Tom Fairgrieve	tff@cs.utoronto.ca <a href="http://www.cs.utoronto.ca/~tff/">http://www.cs.utoronto.ca/~tff/</a>	BA4230 416-978-3965
W. James MacLean	maclean@eecg.utoronto.ca <a href="http://www.eecg.utoronto.ca/~maclean/">http://www.eecg.utoronto.ca/~maclean/</a>	EA314 416-946-7285

### Grading

The grade in the course will be based on five tutorial quizzes, one midterm test, eight labs, and the final examination. The midterm will be cover all material up to a point that will be specified in class. The final examination will be cover all material in the course. The composition of the final grade is as follows:

Item	Weight
Labs	10%
Quizzes	15%
Midterm	25%
Exam	50%

Quizzes will be held in your tutorial rooms. The midterm will be held on November 6th from 6:00pm to 8:00pm. Your assigned room for writing the midterm will be announced in class. Neither calculators nor any other aids are allowed for either the midterm test or the final examination. Note that this means the final exam is **closed book**.

### Course Web Site

The home page for this course can be found at <http://ccnet3.utoronto.ca/20079/aps105h1f/>. Class announcements and all handouts, labs, and other information will be accessible there. Information will be updated on a regular basis, so check the website regularly. You may also verify your marks for labs, quizzes, and tests on the website. The web site will also serve as home for a bulletin board where you can post questions about course content for discussion with your fellow students. *You must not post partial or complete solutions to assignments, or detailed descriptions of solutions, on the course website—only TAs and instructors may do this.* If you contribute to the bulletin board, remember that it is a public forum; you should not be cluttering it with material that is of unrelated to the course, and you should be careful that the content of your posts is not offensive. Be polite and professional!

## Lecture Schedule

Lectures begin at ten minutes past the hour and end at the hour. In the schedule, MC denotes the Mechanical Engineering Building and BA denotes the Bahen Centre for Information Technology.

Section	Instructor	Schedule
LEC0101	Fairgrieve	T10 – MC252 W15 – BA1190 F10 – BA1180
LEC0102	Aboelsaadat	M9 – MC252 T10 – BA1170 F10 – MC252
LEC0103	Fairgrieve	M15 – MC252 T9 – MC252 F14 – BA1130
LEC0104	MacLean	T11 – BA1170 W16 – BA1180 F14 – BA1170
All (Extra Help)	Ahmed Abdelkhalek	T16 – MS2158
		(except T17 in MC102 on Sept 11 only)

The following schedule will give you an idea of the material to be presented, but it may be altered slightly. Any revisions will be announced in class and posted on the course web site.

Week	Date	Lecture & Tutorial Topic	Lab Assignment	Quiz
0	Sep 7	Introduction	None	
1	Sep 10	Variables, Assignments & Expressions	Lab 0: Introduction to UNIX/Linux	
2	Sep 17	Conditionals & Loops	Lab 1: Expressions & I/O	
3	Sep 24	Functions	Lab 2: Conditionals & Loops	Q1
4	Oct 1	Functions, Stream I/O	Lab 3: Functions	
5	Oct 8 Oct 8	Arrays <i>Thanksgiving Holiday</i>	None	Q2
6	Oct 15	Strings	Lab 4: Stream I/O	
7	Oct 22	Searching & Sorting	Lab 5: Arrays & Strings	Q3
8	Oct 29	Recursion	Lab 6: Searching & Sorting	
9	Nov 5 Nov 6	Pointers & Dynamic Arrays <i>Midterm</i>	None	
10	Nov 12	Structures	Lab 7: Recursion	Q4
11	Nov 19	Linked Lists	None	Q5
12	Nov 26	Review	Lab 8: Structures & Linked Lists	
13	Dec 3 Dec 5	Review <i>Last Day of Lectures</i>	None	

A tentative list of quiz topics is :

Q1	- Basic C++ Programming,
Q2	- Conditionals & Loops,
Q3	- Functions,
Q4	- Searching, Sorting & Arrays, and
Q5	- Pointers.

In addition to the material listed above, we will present material on the UNIX/Linux operating system, with emphasis on shells, the file system, and utilities for software development. This material will be tested on the midterm and the final exam. A more detailed list may be found in the course syllabus on the course website.

## Lab Assignments

Labs will consist of programming assignments using the C++ programming language. The first meeting is during the week starting Monday Sept 10th. A handout describing each assignment will be available on the course web site the week prior to the assignment's due date (at the latest). There will be no hard copies handed out in the lectures, so make sure you check the course web site for new lab assignment handouts. In your timetable, you have been assigned one two-hour period for completing and presenting your lab assignments. The schedule that follows shows the times for the lab sections (The letter R in the schedule denotes thuRsdays). All labs are held in Room SF1013 in the Sandford Fleming Building. Since the lab rooms are booked to capacity, please attend only the lab section for which you are scheduled. Teaching assistants (TAs) will be available to answer any questions and offer help in the lab. Your labs are always due in the lab session that you have been assigned to. The lab assignments will likely take some time; it is important to start as early as possible on your assignments and to avoid procrastination.

Section	Schedule
PRA0101	R14-16
PRA0102	T12-14
PRA0103	T10-12
PRA0104	M14-16

In addition to your allocated time in SF1013, you are also permitted to work at other times in the Engineering Computing Facility (ECF) labs located in SF1012 and SF1106. You may also work on your C++ assignments on your own computer, but for marking purposes we will always run your programs on ECF. Always test your programs on ECF before the lab marking session begins. If a program does not work on ECF, it is considered incorrect and you will not get full marks. There will be no exceptions to this rule.

All labs except for Lab 0 will be marked. Although the marks allocated to lab assignments do not contribute significantly to your final grade, you should be sure to try hard to finish each one. The experience gained in doing the labs is probably the most important factor in determining your success on the quizzes, the midterm test and the examination. The test and exam form the bulk of your final grade.

The labs will be marked during the lab session by Teaching Assistants. At the start of each lab, the Teaching Assistants will visit the workstation of each student to offer one-on-one assistance. Once less than 30 minutes time remains in the lab period, the Teaching Assistants will begin to evaluate each student's understanding of the lab by doing some of: asking you a few questions about the lab, reviewing your program, or running your program. You will be assigned a grade between 0 and 2 for each lab. The marking is both objective and subjective. The marking is objective in that if your program generates correct output for a set of input that the TA provides you, you will get some of the credit for your work. The marking is subjective in that if you fail to show sufficient understanding of the lab, your mark may be reduced, since it is likely that you do not understand your program well. Program style will be evaluated.

The lab sessions are mandatory. You have to attend each of the sessions. If you have completed the lab assignment before the marking time starts (including before arriving at the lab), you may ask for your lab to be marked so that you may leave early.

**Important note:** During the lab sessions, do not exchange knowledge or information of any questions that the Teaching Assistants may be asking. These questions should be treated as exam questions. Failure to observe this policy will be treated as cheating.

## Accessing ECF From Your Own Computer

You can access ECF from home using your own Internet Service Provider (ISP) or the UTORdial facility. All U of T students get 20 hours of free UTORdial access per semester. Even if you have another ISP, it is very convenient to have a UTORdial account in case your ISP, or the network between your ISP and U of T, is having trouble. Additional 20-hour units (tokens) may be purchased. Information about internet access is available at:

<http://www.utoronto.ca/ic/internetaccess/>

The following link provides more information about the UTOReMail facility which gives you a central U of T e-mail address that you can keep for life (ECF e-mail accounts will disappear after you graduate):

<http://www.utoronto.ca/ic/utormail/>

Information on wireless access (for laptops) may be found at:

<http://wireless.utoronto.ca/>

Once your computer is connected to the Internet, you can use a program named `ssh` to open up a terminal session to ECF. Information about `ssh` is available on the ECF web site (near the bottom):

<http://www.ecf.utoronto.ca/ecf/>

Also, if you want to transfer files between your computer and ECF then you can do so using `scp` or some other file transfer program. Free copies of `scp` and `ssh` are available on the web. For more information on `ssh` and `scp`, see

<http://www.openssh.org/windows.html>

Please be careful when transferring files; even experienced people have inadvertently ruined their files by sending them in the wrong direction! Always keep backup copies. The course web site has further information on means of accessing ECF from home.

Another alternative for doing assignments at home is to install Linux on your home machine, or even use a *Live CD* version. Ubuntu, a version of Linux that is particularly easy to install and use, is available at: <http://www.ubuntu.com/>. Many varieties of Linux can be downloaded for free from the internet.

## Cheating

**You may be requested, without advance notice, to submit the source code for your lab assignments so that we may check for plagiarism.**

In developing solutions to labs, you may wish to discuss your approach to the solution with fellow students. *This is fine as long as the final solution is yours alone.* If you discuss some aspect of a lab with another student, neither of you should take any notes. You should never copy your lab solutions or let another student copy yours, not even a portion of them. Failure to adhere to this rule is considered to be cheating — *by both parties involved.* There are many signs to indicate that a solution has been copied and we have sophisticated cheating detection programs to look for these signs. We will be comparing assignments to look for copying — if you cheat, you will very likely be caught. To be sure that you avoid the possibility of another student copying your program, you should never let another student look at your program. Do not show another student your program and do not leave a printed copy of your program where others may be able to use it.

The University takes cheating of any kind very seriously. Possible penalties for cheating include: a mark of zero on the assignment, a mark of zero plus a further mark penalty, a mark of zero in the course, loss of computer privileges, annotations on your transcript, or even expulsion from the University. The penalties are chosen to fit the offence. It is simply not worthwhile to take the risk, no matter how desperate you may feel. It is far better to submit an incomplete lab or no lab at all rather than take the chance of incurring the kinds of penalties that the Department may impose.

## Tutorials

In your timetable, you have been assigned a one-hour weekly tutorial. The tutorial schedule is shown below. In the schedule, GB denotes the Galbraith Building and WB denotes the Wallberg Building.

Section	Time	Room
TUT0101	W14	GB412
TUT0102	W14	WB119
TUT0103	W14	WB342
TUT0104	W14	WB219
TUT0105	W15	WB342
TUT0106	W15	WB219
TUT0107	F15	WB119
TUT0108	F15	WB219

Tutorials are conducted by TAs. Attendance is mandatory, since short quizzes are held from time to time in tutorials. It is essential that you interact with the TAs and ask them to explain material you do not understand, either from the labs or lectures. If you don't ask questions, the TAs will have little to say. No new material will be presented in the tutorial sessions, but additional examples may be used to clarify points made in the lectures.

## Text & References

The primary textbook for this course is **Problem Solving with C++, 6th Edition** by Walter Savitch (Addison Wesley, 2007). This text is available from Engineering Stores in the basement of Sandford Fleming and the University of Toronto Bookstore.

If you wish to consult other books, that is fine. **The C Programming Language** by Kernighan and Ritchie (Prentice Hall), while not intended as a textbook, has launched many successful programming careers.

There are many other C++ textbooks available but not all of them are suitable as introductory texts or references. Do not blindly buy a book just because it says C or C++ on the cover — some of them are terrible! As well as purchasing the text, all students should go through the **Getting Started with UNIX Guide** for help when getting started using the ECF computers in the laboratory. This document is available online at <http://www.ecf.utoronto.ca/ecf/docs/>. There are also other useful documents there. Some of the information on this site is a bit out of date but the essential features should still be correct.

## Getting Help

Although you may never have needed extra help in school before, you may well find that you need assistance this year. If you find that you are having difficulty, *seek help right away*. The course moves very quickly and many of the ideas are cumulative so that a small early problem that is left unsolved can quickly grow into a very large, unmanageable one. Even if you have heard that it is hard to find help at university, the truth is that help is readily available in a number of forms.

- **Self Help**

The best students are ones that are not afraid to experiment. If you don't know how something works, write a short program and try it out! If you are still puzzled, post your question to the bulletin board or bring it to a tutorial, lecture, or office hour.

- **Instructor Office Hours**

All instructors schedule regular office hours when they will be available for extra help. To get help from an instructor, either go to the instructor's office during a scheduled office hour or, if that is not convenient, speak to the instructor after class or send the instructor an e-mail to set up an appointment. **This is an excellent way to get extra help with the course material, please take advantage of it!**

- **Tutorials**

Although TAs will be given topics for discussion in tutorials, questions from students are more than welcome. Questions should be of a type that might be of interest to the class rather than something that is likely to be of concern only to the questioner.

- **Special Sessions**

Every Tuesday, there will be a completely optional extra class held in MC102 at 16 h. This session is open to all students but it is intended primarily for those who have little or no prior experience in computer science. No new material will be covered in these sessions; they will simply attempt to amplify or clarify things that have been dealt with elsewhere.

- **Labs**

The labs are there for you to get some very detailed, personal help from the TAs. If you find yourself stuck on a problem in a lab, do not hesitate to call on a TA.

- **The Course Bulletin Board**

If you would like clarifications or explanations of course material, you do not have to wait until your next lecture! You may check the bulletin board section of the course web site — someone else may have asked the same question already. The bulletin board is intended to be a discussion forum for students, and although instructors will monitor the bulletin board they will not post responses to all questions. Fellow students may help you out by replying to your questions. Even if you do not have a question to ask, it is a good idea to look at the bulletin board frequently. You may see a question and/or a response there that will make you realize that there is some aspect of the course material that you had never considered. *While it is OK to discuss assignments in general terms on the bulletin board, you must not post partial or complete solutions to assignments, or detailed descriptions of solutions. Only TAs and instructors may post such material.*

If you are considering posting a question to the bulletin board, you should do two things before doing so. The first thing is to read the bulletin board to see if your question has already been asked and answered. Failure to do so may waste the time of other readers of the bulletin board and put you at risk of receiving an unpleasant/sarcastic/humiliating reply. The second thing that you should do is make sure that your question is as clear as you can make it. Sending a note saying “My program does not work” will not elicit much sympathy since it is too vague. In addition, make the title of your posting as clear as possible. “Lab 4 - Required Output Format” is better than “Lab Problem”. Note that the bulletin board should not be used for personal questions. If you have a problem of a personal nature (e.g. what to do about missing a tutorial quiz because of illness), please send e-mail to your instructor. Also, as we said earlier, you should never make postings to the bulletin board that others might consider to be offensive. Please do not make posts unrelated to the course, and under no circumstances will profanity be tolerated. Finally, please use proper, professional English, not MSN speak.

## **Exercises — Practise! Practise! Practise!**

Time and time again, experience has shown that practise is the only way to learn programming. If you do only the assigned laboratory exercises of this course, you may not develop your programming skills adequately to earn a decent mark in this course nor to sustain you in the requirements of subsequent computing courses. There are many exercises in the text — both non-programming questions and programming problems. Try as many of these as you can. Instructors and TAs will be happy to help you if you are having difficulty with any of these questions.