

APS 105: Computer Fundamentals — Fall 2000

John Carter, Guy Lemieux, James MacLean

1 General Information and Instructors

APS 105 – Computer Fundamentals – is designed to give you an introduction to computer systems and their role in problem solving in science and engineering, and to provide you with the basics for studies in greater depth in subsequent years. The course consists primarily of lectures and laboratories. Major topics to be covered in the lectures include: digital computer elements and digital computer operation, problem solving, programming, 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 *Java* programming language. Throughout, there will be an emphasis on a programming style that encourages easier understandability, debugging, and maintenance of programs.

John Carter	jcarter@ecf.utoronto.ca SF2104	http://www.ecf.utoronto.ca/~jcarter/ 946-5209
Guy Lemieux	lemieux@ecf.utoronto.ca SF2002A	http://www.ecf.utoronto.ca/~lemieux/ 946-7038
James MacLean	maclean@ecf.utoronto.ca 203 College Rm. 508	http://www.ecf.utoronto.ca/~maclean/ 946-7285

2 Grading

In addition to the lab work, there will be five tutorial quizzes, two midterms and a final exam. The midterms will be cumulative, covering all material up to the specified date. The final exam will be comprehensive, covering all material. The tests are on the following days:

What?	When?	Where?
5 Quizzes	As announced in class	In your tutorial room
Midterm 1	October 2 6–7:30pm	L0101 — MS3153 L0102 — SF3201/2 L0103 — SF1013
Midterm 2	November 13 6–8pm	L0101 — MS3153 L0102 — SF3201/2 L0103 — SF1013

The composition of the final grade is as follows:

Lab Assignments	10%
Lab Project	5%
5 Tutorial Quizzes	5%
Midterm 1	10%
Midterm 2	20%
Final Exam	50%

No calculators and no aids are allowed for the midterms or the final exam.

3 Home Page and Newsgroup

The home page and newsgroup for this course can be found at

web page <http://www.ecf.utoronto.ca/~aps105w/>
 newsgroup ut.ecf.aps105

Class announcements and all handouts, labs, and other info will be accessible there.

There is also a course newsgroup, which you should use to ask questions about anything else you think is course related. The newsgroup name is generally accessible only from ECF. Although you may be able to read this newsgroup from other news servers, **messages posted from those servers will probably be lost and never come back to ECF.** *Always check that your news posting did go through by reading the newsgroup afterwards.* It may take a few minutes for your article to show up.

One additional news server you can use is the `netnews.utoronto.ca` news server, `news://netnews.utoronto.ca/ut.ecf.aps105`

You should have access to this server from your UTORDial account (see Section 7 below), and other computers on campus, such as your connection in residence or the library.

4 Lecture Schedule

Instructor	Section	Monday	Tuesday	Wednesday
John Carter	L0103	11–12, MC252	10–11, MC254	9–10, MC252
Guy Lemieux	L0102	4–5, MC252	3–4, SS2102	11–12, SF1105
James MacLean	L0101	2–3, MC252	10–11, MC252	9–10, SF1101

2000

Sep							Oct							Nov							Dec						
S	M	Tu	W	Th	F	S	S	M	Tu	W	Th	F	S	S	M	Tu	W	Th	F	S	S	M	Tu	W	Th	F	S
					1	2	1	2	3	4	5	6	7				1	2	3	4						1	2
3	4	5	6	7	8	9	8	9	10	11	12	13	14	5	6	7	8	9	10	11	3	4	5	6	7	8	9
10	11	12	13	14	15	16	15	16	17	18	19	20	21	12	13	14	15	16	17	18	10	11	12	13	14	15	16
17	18	19	20	21	22	23	22	23	24	25	26	27	28	19	20	21	22	23	24	25	17	18	19	20	21	22	23
24	25	26	27	28	29	30	29	30	31						26	27	28	29	30	24	25	26	27	28	29	30	

Week	Date	Lecture and Tutorial Topic	Lab Assignment
1	Sept. 11	Java variables, assignment statements	Lab 0. Intro to Unix
2	Sept. 18	selection and looping	Lab 1. Basic Programming
3	Sept. 25	methods	Lab 2. Loops and Conditionals
4	Oct. 2	arrays, strings	Lab 3. Methods
	Oct. 2	First midterm. 6pm-7:30pm.	
	Oct. 9	<i>Thanksgiving Holiday, the university is closed.</i>	
5	Oct. 9	objects	Lab 4. Arrays
6	Oct. 16	objects, simple linked lists	Lab 5. Objects
7	Oct. 23	developing algorithms, planning	Lab 6. Linked List
8	Oct. 30	searching, sorting	Project Week 0
9	Nov. 6	recursion, quicksort	Lab 7. Searching and Sorting
	Nov. 13	Second midterm. 6pm-8:00pm.	
10	Nov. 13	applets (tentative)	Lab 8. Recursion
11	Nov. 20	more linked lists	Project Week 1
12	Nov. 27	stacks, queues, examples	Project Week 2
13	Dec. 4	binary trees, review	Project Demo
	Dec. 6	<i>Wednesday, last day of classes.</i>	
	Dec. 11	<i>Monday, first day of exams.</i>	

5 Labs and Lab Assignments

5.1 Lab General

The schedule below shows the times for the lab sections. All labs are held in SF1012, SF1106, and GB251. Access to the ECF rooms SF1012 and SF1106 is permitted at any time. However, use of GB251 is restricted to your scheduled lab period only — please use the other two rooms outside of your scheduled lab period.

Instructor	Section	Rooms	Tuesday
John Carter	P0103	SF1012, SF1106, GB251	2–4
Guy Lemieux	P0102	SF1012, SF1106, GB243	10–12
James MacLean	P0101	SF1012, SF1106, GB251	5–7

You must regularly attend your own scheduled lab session. Since the lab rooms are booked to capacity, *please attend only the lab section you are scheduled for*. Teaching assistants will be available to answer any questions and offer help in the lab.

You may work on your Java assignments on your own computer, but your labs will *always* be run and marked on ECF. Always test your programs on ECF. *If it doesn't work on ECF, it is considered incorrect and you will not get full marks. There will be no exceptions to this rule.*

Your lab assignments consist of programming assignments using the Java programming language. A handout describing each assignment will be available on the home page by the week prior to the beginning of the assignment. The lab assignments will take a substantial amount of your time. **It is important to start as early as possible on your assignments and to avoid procrastination. Don't start a lab assignment during the lab in which it's due.** Average students who are new to programming have said that it takes them **6 to 10 hours to complete a lab**.

Although the assignments do not contribute significantly to your final grade, you should be sure to finish each assignment. The second midterm and the final exam contributes substantially to

your grade. You will not be able to score well on these tests if you do not independently complete all the assignments.

There will also be a *Lab Project* during the last several weeks. This is an opportunity for you to be creative and implement something interesting. Projects will be decided in consultation with the TAs and the instructors. Project demonstrations will occur during the last lab period of the term.

5.2 Lab Grading, Due Date, Cheating Policy

Your labs are always due by 11:59pm Wednesday. Late assignments will not be marked. You will electronically submit your lab using the `submitaps105f` command on ECF. Details on how to use the command are included in *Lab 0*.

The labs will be graded out of 6, with two components to the mark: functionality (3 marks) and documentation/style (3 marks). Full marks for functionality are given if the program works correctly, less if not. Full marks for documentation are given if variable names are chosen well, and the code is properly commented and indented. Later in the course, good style will include proper design and modularization of your code.

Do not copy your lab solutions, not even a portion of them. There are many signs to indicate a solution has been copied, and we use sophisticated cheating detection programs to look for these signs. We will be comparing all assignments to look for copying — if you cheat, you will be caught. **The penalty for copying is a grade of zero in the lab, and a loss of twice the lab value from your final mark.** Most importantly, you will not score well on your exams, and this will have a very large impact on your grade.

6 Tutorials

Section	Thursday	Friday
T0101	11–12, SF1013	
T0102	11–12, GB405	
T0103	11–12, SF3202	
T0104		11–12, SF3201
T0105		11–12, GB405
T0106		11–12, SF3202
T0107	4–5, SF1013	
T0108	4–5, SF4102	
T0109	4–5, WB219	

Tutorials are scheduled every week. **Attendance is mandatory, since a short quiz may be held by the TAs.** It is essential that you interact with the TAs and ask them to explain material you still don't understand, either from the labs or lectures. If you don't ask questions, the TAs will have little to say. The first few weeks are essential in helping you cope with programming the first time.

No new material will be presented in the tutorial sessions, but additional examples may be used to clarify points made in the lectures.

7 Accessing ECF from Home

You can access ECF from home using your own Internet Service Provider (ISP) or the UTORdial facility.

All UofT 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 UofT, is having trouble (*note*: lately, UofT has had trouble connecting to the outside network on a regular basis). Additional 20 hour units (tokens) cost \$5 each.

The following link provides for more information about UTORdial, as well as the corresponding UTOReMail facility which gives you a central UofT email address that you can keep for life (ECF email accounts will disappear after you graduate):

<http://www.utoronto.ca/welcome.html/maildial/index.html>

Once your home computer is connected to the Internet, you want to use telnet to open up a terminal session to ECF. Also, if you want to transfer files then you should use ftp. Please be careful when using ftp, because even experienced people have inadvertently clobbered their files by sending them in the wrong direction! *Always keep backup copies.*

8 Office Hours

It is best to get a hold of the instructor immediately after lectures or by booking an appointment through email. You should also see the Getting Help section below.

9 Textbooks

The primary textbook for this course is:

Cay Horstmann
Computing Concepts with Java 2 Essentials, 2nd Ed.
John Wiley & Sons, 2000

Please be certain to get the *second* edition of this text – it is significantly easier to follow than the first edition. This text will be available in the University of Toronto Bookstores, and is also available from Discount Textbooks, Chapters, Indigo, and other retailers of computer books.

If you wish to consult other books, that is fine. A number of good alternatives will be listed on the course web page. However, there are a very large number of Java textbooks and not all of them are suitable as introductory texts or references – don't blindly buy a book because it says Java on the cover.

In addition, John Carter will be offering a set of notes specifically tailored to APS105 course material. These notes will be very helpful, especially for first-time programmers. They may be purchased directly from him at a cost of \$20.

All students should go through the *Getting Started with UNIX* guide to be able to use the Engineering Computing Facility computers in the laboratory. This is available online at

There are also other useful documents there.

In addition to the above resources, every student in the course is expected to understand a number of basic concepts in computer hardware, computer software and computer science, which will be covered in the lectures, but will not necessarily be found in the textbook.

10 Getting Help

The labs are there for you to get some very detailed, personal help from the TAs. However, if you would like clarifications or explanations of laboratory materials, don't wait until your lab period! You should first check the course webpage and newsgroup — someone else may have asked the same question already.

The best students are ones that are not afraid to be experimental. 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 newsgroup. To give you the quickest response, all of the TAs and instructors will be monitoring the newsgroup. Fellow students may help you out by replying to your questions as well. (If your question is of a more personal nature, please send email to one of the TAs or an instructor.)

If you run into some code that does not work, you have tried everything you know, and you are really stuck, sending mail to someone saying "My program does not work" will not elicit much sympathy. You should narrow down the problem by trying this:

- Try to create the smallest program that still has the problem. Something of the order of 20 lines or less is ideal. You can do this by deleting lines in your program until you isolate the problem area.
- See if you can make this small example work.
- Add `if` and `print` statements to test some of your assumptions at strategic locations in your program.
- If you are still stuck, then mail the program to a TA with sample inputs, what you expect, and what you see. Explain what you want to do, and what you have tried.

11 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 will not develop your programming skills adequately to earn a decent mark in this course nor to sustain you in the requirements of the second year computing courses. There are many additional programming exercises in the textbook and John Carter's notes.