APS 105F – Computer Fundamentals

Paul Chow

Fall 1997

1 General Information

Important Note: The fall version of APS105 will assume that everyone has some programming experience. If this is not the case, then you should be in the Spring section, and take the remedial courses that are being offered.

APS 105S — 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 of lectures, laboratories, and tutorials. Major topics to be covered in the lectures include: Problem solving; Algorithms; Data Structures; Programming in *Turing*, *C*, and some *C++* or *Java* to introduce the ideas of object-oriented programming. There will be brief introductions to Computer Organization (how a computer is put together and works inside), and Operating Systems (the software that controls the computer). The laboratories will provide you with an opportunity to exercise concepts introduced in the lectures through a number of programming assignments using the *Turing*, *C*, and *C++* or *Java* programming language. Throughout, there will be an emphasis on a programming style that encourages easier understandability, debugging, and maintenance of programs.

2 Home Page

There is (will be) a home page for this course. It can be found by going to

http://www.ecf.toronto.edu/apsc/courses/

and clicking on the link to APS105F or by going through my home page (see instructor info). The most recent versions of this handout, other handouts, and the labs, will be accessible there. Watch the home page for updates on test dates and locations, and other news for the class.

3 Instructor

Professor Paul Chow
LP484A
978-2402
pc@eecg.toronto.edu
http://www.eecg.toronto.edu/~pc

4 Office Hours

It works best to get me after lectures or make an appointment. You can also send me email.

You should also see the Getting Help section below.

5 Lectures

Monday	Wednesday	Thursday
11–12	9–10	11–12
MC102	MC102	SF1105

6 Labs and Lab Assignments

The schedule below shows the times for the lab sections. While lab access and use are possible any time, you *must* regularly attend your own scheduled lab session. We will not accept your assignment unless it is your day or by **permission of the instructor**. Teaching assistants will be available to answer any questions and offer help at those sessions, and will assign you a grade based on your progress in each lab session. If you have the assignment working before the lab session, then you only need to attend long enough to get a grade assigned.

Grades for the labs are assigned as shown in the table below.

Wednesday P0101	Thursday P0102
2–4	2–4
SF1012	SF1012
SF1106	SF1106
EA109	EA109

1	Attended lab
2	Program is written, has major problems
3	Program has minor problems
4	Program works for all cases
5	Above and beyond call of duty: extra features, be creative

The lab assignments consist of programming assignments using the Turing and C programming languages. 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.**

Although the assignments do not contribute significantly to your final grade, you must work and finish each assignment. The midterm, which contributes substantially your grade, will have a lab component based largely on these assignments. It is unlikely you will be able to score well on the midterm if you do not work independently on and finish the assignments.

There will also be a *Lab Project* that will last two periods. This is an opportunity for you to be creative and implement something interesting. Projects will be decided in consultation with the TAs. Demonstrations will occur during the second period.

The Lab schedule will be as follows this term:

Assignment:	1	2	3	4	5	6
P0101 Due:	Sept 10	Sept 17	Sept 24	Oct 1	Oct 8	Oct 15
P0102 Due:	Sept 11	Sept 18	Sept 25	Oct 2	Oct 9	Oct 16
Assignment:	7	Midterm	8	9	Proj I	Proj II
P0101 Due:	Oct 22	Oct 29	Nov 5	Nov 12	Nov 19	Nov 26
P0101 Due:	Oct 23	Oct 30	Nov 6	Nov 13	Nov 20	Nov 27

7 Tutorials

Tutorials are scheduled for Monday from 2–4pm in room SF1105. However, they will be held **only when necessary**, as announced in class, or by request. No new material will be presented in the tutorial sessions. They are used to deal with difficulties experienced in in understanding key concepts or programming.

8 Textbooks

For those students **who already have programming experience**, there is no required textbook.

Those **students with little or no programming experience** require the text:

> Elliot B. Koffman, Richard C. Holt, and Chrysanne Dimarco, *Turing - Problem Solving and Program Design*, Addison-Wesley, Don Mills, 1994. ISBN-201-42640-4

All students will need a *Turing* language reference book. The book by Koffman, Holt and Dimarco is quite suitable for this purpose, but a cheaper alternative for **experienced programmers** is:

> R.C. Holt, *Turing Reference Manual*, fifth edition, HSA, Toronto, 1994.
>
> Earlier editions are generally okay, but you may miss a few minor things.
>
> This book is available with or without software. You do not need to buy the software unless you wish to run Turing on your own personal computer.

For *C* programming, there are many, many choices. For **experts** a good reference manual is sufficient:

> Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language*, second edition, Prentice Hall.

A gentler introduction that includes a section on *C++* can be found in:

> H.M. Deitel and P.J. Deitel, *C: How to Program*, second edition, Prentice Hall, 1994.

The above books are available at the University Bookstores. There will also be an *OOT Software Sale*. See the section below.

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 http://www.ecf.toronto.edu/ecf/docs/. There are also other useful documents there.

In addition to the above books, 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 recommended books above. The lecture material should be sufficient. If you want supplementary reading, you can consider:

Computer Structures Carl Hamacher, Zvonko G. Vranesic and Safwat G. Zaky, *Computer Organization*, McGraw Hill, Toronto, 1996. Fourth edition: Chapter 1.

Data Structures J. Glenn Brookshear, *Computer Science - An Overview*, Benjamin Cummings, Redwood City, Calif. Third or fourth edition: Chapter 7.

Algorithms J. Glenn Brookshear, *Computer Science - An Overview*, Benjamin Cummings, Redwood City, Calif. Third or fourth edition: Chapter 4.

Les Goldschlager and Andrew Lister, *Computer Science: A Modern Introduction*, Prentice-Hall International, London, England, 1988. Second edition: Chapter 2.

Operating Systems J. Glenn Brookshear, *Computer Science - An Overview*, Benjamin Cummings, Redwood City, Calif. Third or fourth edition: Chapter 3.

Les Goldschlager and Andrew Lister, *Computer Science: A Modern Introduction*, Prentice-Hall International, London, England, 1988. Second edition: Section 5.5.

9 Grading

There will be one midterm test and a final exam. The midterm will contain two components: a lab component where you will be asked to write some programs based on the lab assignments and submit them, and a written component based on the material covered by the lectures. The final exam will be comprehensive, covering all material. The midterm will be scheduled during one of your lab periods.

Who?	When?	Where?
P0101	Oct. 29, 2–4pm	SF1012, SF1106, EA109
P0102	Oct. 30, 2–4pm	SF1012, SF1106, EA109

The composition of the final grade is as follows:

Lab Assignments	10%
Lab Project	15%
Midterm	20%
Final Exam	55%

The exam and midterm are **No Calculators** and **No Aids (Type A)**.

10 Getting Help

If you would like clarifications or explanations of laboratory materials, the first sources of help are the Teaching Assistants (TAs) in the laboratory. Sending email to one of the TAs should also elicit some response. Otherwise, feel free to get in touch with the 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. Try 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.
- If you are still stuck, then mail the program to me or 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–Practice! Practice!

Practice 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 programming exercises in Koffman and Deitel. For many of them, it does not matter what language you use. If you have those texts, you should do enough of them to make yourself confident in programming.

12 OOT Software Sale

HSA Inc. will be having a sale giving you a chance to buy the OOT software for much better than at the Computer Shop (I'm told). Time and location are on the form below. **You need to fill it out and take it with you.**

Students who already have legal software can bring their disk and Reference Manual as proof of purchase and get their software registered. Students must complete a form and bring it along with their payment (or materials to be registered).

Please keep in mind that students can continue to purchase the software package throughout the school year at the University of Toronto Computer Shop at the regular price.


```
                 OOT SOFTWARE FOR COMPUTER SCIENCE COURSES

Again this term we will be offering an opportunity for students to
purchase discounted OOTuring packages for students with home computers.

Student should be aware that this package is for personal/individual use
only (one per student).  Resale is not permitted.

The date and locations are as follows:

Downtown:  Monday September 15, 1997 203 College St. Suite 305 10:00am - 5:00pm

*******************************************************************************


Student Name:                                     CS Course:  APS105F


Instructor's Name: Paul Chow


Turing Registration Number :


(1) WinOOT software                          $35.00
(2) DOS OOT software                         $35.00
(3) MacOOT software                          $35.00
(4) WinOOT software & User's Manual          $45.00
(5) DOS OOT software & User's Manual         $45.00
(6) MacOOT software & User's Manual          $45.00
(7) OOT User's Manual Only                   $10.00
(8) Object Oriented Turing Package           $90.00
  (WinOOT, DOS OOT, User's Manual, Reference Manual)

Upgrades:
(1) upgrade from Turing to OOT (software only)    $25.00
(2) upgrade from Turing to OOT (full package)     $65.00
Students requesting upgrades must bring their disks.
```