APS 105: Computer Fundamentals – Winter 2017 Basic Course Information

Welcome! Computer software has become one of the most important engineering tools ever created. Even in the last few years, software has become essential in almost every human endeavour, whether it is music, design, healthcare, business or entertainment. The goal of this course to introduce you to the basics of computer software – and using it to solve problems – and to be your first taste of what engineering design really is all about.

It will also to provide you with the basics for studies of software in greater depth in later courses. You will learn how to make high-quality software through participating in lectures, but most importantly by the work you do for the labs. The main topics to be covered are: the representation of information, the development of good programming techniques, program organization, systematic ways of finding and fixing errors in programs, algorithms, and data structures. We will focus on the C programming language, but the concepts taught apply to most languages.

Instructors

Course coordinator: Jonathan Rose
Lecture Sections 3 and 4
Email: Jonathan.Rose@ece.utoronto.ca
Web: `http://www.eecg.toronto.edu/~jayar/`
Office: EA 319

Instructor: Baochun Li
Lecture Section 2
Email: bli@ece.toronto.edu
Web: `http://iqua.ece.toronto.edu/bli`
Office: BA 4118

Instructor: Michael Stumm
Lecture Section 1
Email: stumm@ece.utoronto.ca
Web: `http://www.eecg.toronto.edu/~stumm/`

Office: SF 2001

Office Hours: Each instructor will convey their individual office hours during class.

Grading

The grade in the course will be based on ten labs, the midterm examination, and the final examination. The midterm examination will cover all material up to a point that will be specified in class. The final examination will be comprehensive, covering all material, but with the emphasis on material covered after the midterm examination. The composition of the final grade is as follows:

Item	Weight
Labs	25%
Midterm	25%
Final Exam	50%

The midterm examination will be held on Tuesday February 28, 2017, at 1:00pm. Calculators are not allowed in the midterm examination, or the final examination.

Course Web Site

The web site for this course can be found on BlackBoard, accessible from the University of Toronto Portal `portal.utoronto.ca`. Look for the course title *Computer Fundamentals*. All class announcements, and handouts - including labs, and solutions will be accessible there. We will use a different web site, Piazza.com, to serve as a home for a course discussion board where you can ask questions about course content and assignments.

Schedule

Lectures and tutorials begin at ten minutes past the hour and end at the hour.

Time	Monday	Tuesday	Wednesday	Thursday	Friday
9:00	TUT 07		**Plenary Lecture** (MS 2158)		PRA 02 PRA 03
10:00			TUT 01		PRA 02 PRA 03
11:00	LEC 04		LEC 04	LEC 04	
12:00	LEC 02 PRA 04 TUT 06		LEC 02 TUT 03 TUT 08 TUT 09		PRA 01
1:00	PRA 04				PRA 01
2:00					LEC 02
3:00		LEC 01		LEC 01	LEC 01
4:00	LEC 03	TUT 05		LEC 03	
5:00	TUT 02 TUT 04	LEC 03			

Lab Assignments

The crucial part of this course is to learn programming by actually doing it. We provide strong TA support in the labs to both help you, and to make sure you're learning the key things. You will be expected to work on your labs both on your own time, on your own computer or using the Engineering Computing Facility computers in room SF 1012, 1013 and 1106.

There are ten labs in this course (including lab 0), and the lab schedule is given in Table 1, below. Every lab (except Lab 0) has grades associated with it, with the earlier labs worth a smaller fraction of the lab mark, and later labs worth more as shown in the table. It is essential to learn all of the earlier material as the later content will be based on it!

A handout describing each assignment will be available on the course web site roughly two weeks prior to the week in which it is due. The grade for the lab will consist of two parts - some of the grade will

Friday/Monday PRA 1-2-3/PRA 4	Lab	Portion of Grade
January 13/16	Lab 0	–
January 20/23	Lab 1	1%
January 27/30	Lab 2	1%
February 3/6	Lab 3	2%
February 10/13	Lab 4	2%
February 17/27	No Lab this week which spans reading week	–
March 3/6	Lab 5	3%
March 10/13	Lab 6	4%
March 17/20	Lab 7	4%
March 24/27	No Lab due; Work on Lab 8!	–
March 31/April 3	Lab 8	4%
April 7/10	Lab 9	4%
Total		25%

Table 1: Lab Schedule and Portion of Grade for Each Lab

be assigned by your Teaching Assistant during your specific lab period, and some of it will be done by software that runs your program and checks its output. Each lab will be due by the end of your specific lab period. Late submissions will receive a mark of zero.

These lab assignments will likely take a substantial amount of your time; it is important to start as early as possible on your assignments. **You are free to discuss your approach with your fellow students; however you may not copy all or part of someone else's assignment solution. Furthermore, under the University of Toronto code of conduct, a person who supplies an assignment to be copied will be penalized in the same way. We will use software to detect copying that is quite sophisticated and so is difficult to defeat.**

In your timetable, you have been assigned a two-hour period for working on your lab assignments. The lab will be held in room SF1013 (and possibly SF 1012, next door) in the Sandford Fleming Building. There are roughly 12 students to be permanently assigned to one Teaching Assistant (TA). That TA will do all of the in-person grading for all of your labs, which will consist of marking both the style of your programs, and a few related questions to test your understanding of the material.

In addition to your allocated lab time, you may work in the ECF labs located in SF1012, SF1013, and SF1106 when they are not booked by another group. You may also work on your assignments on your own computer, but for marking purposes we will always run your programs on ECF. *Always* test your programs on ECF computers before submitting them. If a program does not work on ECF, it may be considered incorrect when the automarker checks your program with some of the test cases. No exceptions to this rule will be made.

The total amount of marks for the labs contribute 25% to your final grade, with earlier labs counting for less. However, you should not skip any labs, as the later more complex labs rely on the basic knowledge learned in the earlier ones.

Using Lab Software At Home - NetBeans

You will be using a software tool that makes it easier to develop software itself – it makes sense that software developers have created many software tools to help in this process! We will be using the **NetBeans** tool, which allows you to create and edit your software, *compile* it (you'll learn what that means) and then *execute*

it. NetBeans is available on the computers in the Engineering Computing Facility that you'll be using in lab, and you can do all of your preparation work (prior to the lab period) on those computers.

We will also provide you with a way to run this tool on your own computer. This will be done with what is called a 'Virtual Machine' that runs on your laptop/computer. The first lab will give you instructions on how to install the needed software on your own computer.

Tutorials

You will have a single one hour tutorial associated with this course. New this year, the tutorials will be making extensive use of the online textbook, so you will need to bring a laptop computer to the tutorials. If you do not have a laptop, please speak with your instructor. Tutorials will begin in the week **January 16, 2017**.

Mandatory Plenary Lecture/Hands-On Teaching

In your schedule, you may notice that it appears to say you have a lab (PRA 010X, X=1,2,3 or 4) on Wednesday morning from 9am-10am, and is located in MS 2158. This is **actually not a lab**, but a separate special seminar in which we will do two things: 1) Bring in guest speakers to talk about novel, interesting, and important uses of software and 2) Insightful hands-on sessions providing a different perspective on the software creation process for new students. The guest speakers are a way for you to understand where software might take you; the second is to touch the real issues associated with learning software for the first time.
We plan to have speakers from industry and research, including the major companies (Google, Facebook) and small startups, as well as interesting new activities in research. These lectures are mandatory, and some of the questions on the midterm and final will relate to topics brought by the speakers. The first plenary lecture will be on Wednesday January 11th from 9am-10am in the Medical Sciences Building room 2158. This room is located a little north of the Medical Sciences Building, in the south-east corner of King's College Circle.

Textbooks

The **required** textbook for this course is:

NEW this year, we are using a textbook that is quite different than previous years. This textbook is online, and one of its great features is that you can watch programs as they execute, in a very intuitive way, and the website itself allows you to write and run simple programs, very easily. It's cost is $USD 63; you'll need a credit card to purchase it online. Note that you can also have the textbook print into a PDF, but that the PDF will not have the interactive features.
To access this book, do the following:

1. Using your usual web browser, go to the following web address: `http://zybooks.com/`
2. Create an Account by Clicking on the 'Create Account' link on the top right of the page, and following those instructions.
3. Once registered, click on the the 'Take me to my Zybooks Library' link.
4. There you will see a page that includes: 'Are you in a class? Enter your zyBook code.' Use this code: **UTORONTOAPS105RoseSpring2017**
5. Spend 15 minutes or so learning how to move around the Zybooks website. Read Section 1.1 of the text and click on 'Run' for the first program shown in that section.

You may also find that a previous textbook used in this course is helpful (this text is not required):

> An Introduction to Computer Science Using C, Second Edition
> Author: John Carter
> McGraw Hill Custom Publishing
> ISBN 1259015262

There may be some of these available in the Engineering Stores in the basement of Sandford Fleming, and in the University of Toronto Bookstore.

This text is also available as an e-book from McGraw Hill. The book can be viewed on a web page, or downloaded onto any Windows or Mac Computer, or Android device, or iOS device, after installing the appropriate application. You can purchase the e-book at University of Toronto bookstores, and by going to this website, and clicking on the book listed as "STG APS105 CARTER 2E EBK" for this course:

`http://uoftbookstore.com/textbooks/access_codes.asp`

A few notes - this site is not terribly intuitive - to purchase, just click on this book, and follow the prompts, which will ask you to create a *separate* account from the main bookstore site. This will give you a code that you need to take to *another* site (the publisher's) on which you use that code. See the following website for information on the application software that runs the e-book:

`http://support.vitalsource.com`

Course Discussion Board

We will be using the Piazza website for the course online discussion board. It will be monitored by instructors and TAs, but we encourage everyone to provide answers in a timely manner to each other. If you were not already signed up to the Piazza website, please go to the following website and signup to this separate website, as soon as possible:

`https://piazza.com/utoronto.ca/winter2017/aps105/home`

If you would like clarifications or explanations of laboratory assignments, do not wait until your scheduled lab section. You should first check the course discussion board on the Piazza web site – someone else may have asked the same question already. Fellow students may also help you out by replying to your questions. Even if you do not have a question about a lab, it is a good idea to look at the course discussion board from time to time. You may see a question or a response there that will make you realize that there is some feature of the lab that you had never considered.
If you are considering posting a question to the discussion board, you should do two things before doing so. First, try to read the current entries to see if your question has already been asked and answered. Second, you should make sure that your question is as clear as you can make it.

Petitions and Re-marking Requests

If you miss a lab assignment, or the midterm examination for a valid reason, you need to submit a petition to the first year office, which will then be sent to your instructor. The petition should be sent to the first year office within 5 days of the event, and include documentation.

If you believe that your lab assignments or midterm examination were not marked correctly, you need to submit a re-marking request form within two weeks, with detailed reasons in writing. The re-marking request form is downloadable from the course web site.

Academic Integrity

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. You should not post any of your assignment questions in a private or public online discussion forum or web site in order to solicit solutions from others. You should not allow any other student to read any portions of your solution, either by sharing a computer screen, or by sharing portions of the code remotely via instant messaging or social networking web sites. Note that, under the University of Toronto code of conduct, a person who supplies an assignment to be copied will be penalized in the same way. We will use software to detect copying that is quite sophisticated and so is difficult to defeat.

e-mail

Please make sure that you check your University of Toronto mail account on a daily basis. You are required to read your email messages that are sent to you at your University of Toronto email address as it may contain important announcements from the First Year Office or the ECE Undergraduate Office.

Getting Help

Although you may never have needed extra help in school before, you may well find that you need assistance in this course. 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 forums.

- **Tutorials**
  During the tutorials this year, we will be making use of the online textbook, to reinforce concepts taught in class by directly working with the. You should bring your laptop computer to the class. If you do not have a laptop computer, please inform your instructor.
- **Office Hours**
  As noted earlier, your instructor is available for extra help during regular office hours. To get help from your instructor, either visit his/her office during a scheduled office hour, or, if that is not convenient, speak to them after class, or send an email to set up an appointment.

Accessibility Needs

The University of Toronto is committed to accessibility. If you require accommodations or have any accessibility concerns, please visit `http://www.studentlife.utoronto.ca/as` as soon as possible.

APS 105 Lecture and Lab Schedule Winter 2017 Last Modified 26-Mar-17

Week #	Date of Monday	Lect #	Letter	Content	Zybook Text Sections	Carter Text Sections	Tutorial	Lab - Friday of this week and Monday of Next	9am Wednesday Plenary Seminar/Handson in MS 2158
1	9/Jan	1	A	Introduction to Course; lab 0 discussion including software demo	1.2, 1.9	1.2		Lab 0 - Login to ECF; assign students to TAs; introduction to Linux and NetBeans	**Prof. Jonathan Rose (UofT) - intro to plenary and Mobile Software**
		2	B	Structure of Computers: CPU, Memory, I/O; A simple program that inputs, computes and outputs.	1.3, 1.4, 1.6,1.7,1.8	1.1, 1.7, 1.8			
		3	C	Process of developing and debugging software; Example program with Fractional Variables, the 'double' type; rules for variable names	1.5,1.6, 1.11, 2.1, 2.2, 2.3 2.16	1.3, 1.4	No Tutorial		
2	16/Jan	4	A	Representation of numbers in binary and effect of number of bits; Types of variables - int and double, Lab 1	2.1, 2.9, 2.5, 2.19	1.5, 1.6		Lab 1 - Review Linux/NetBeans & A Simple Calculation	**Danyao Wang (Google) - Software at Google Scale**
		5	B	Example calculation program with input and output using printf and scanf general form; example calculation; Lab 1	2.10, 2.16	1.8; Appendix B	Tutorial		
		6	C	Calculations: operators +-*/%, example program; math functions, random #s	2.4, 2.7, 2.15	2.1, 2.2, 2.6	1		
3	23/Jan	7	A	Random numbers and Rint(), mixing integers and doubles in calculations, casting; Lab 2 discussion	2.15, 2.7, 2.8	2.3, 2.5, 2.6		Lab 2 - More Complex Calculation	**Prof. Michael Stumm - How Facebook Software is Made**
		8	B	Making decisions example, if, else, conditional expressions, relational operators, control flow; BOOL Type; roulette example	3.1, 3.2, 3.3, 3.6	3.1, 3.2,	Tutorial		
		9	C	Char Type - Comparing characters, complex logical expressions and conditions, lazy evaluation, multiple conditions, dangling else	2.10, 3.4, 3.12, 3.13, 3.14	3.3, 3.4,	2		
4	30/Jan	10	A	Repetition (loops) - example, while loop, flow, do while, example, Lab 3 Discussion	4.1, 4.2, 4.3	4.1, 4.2		Lab 3 - Decisions and Simple Loops	**Prof. Vaughn Betz - An Electrical Engineer's Journey into Software**
		11	B	For loops, flow, example, when to use while, for, dowhile, complex conditions in for, nested loops; increment ++, decrement;	4.4, 4.5, 4.6	4.3, 4.4, 4.5	Tutorial		
		12	C	More nested loop examples; **Functions**, motivation and example with parameters, and prototypes	5.1, 5.2, 5.3, 5.4	4.6,5.1, 5.2	3		
5	6/Feb	13	A	Functions, sending information to functions (parameters/arguments;) call by value. Variable Scope in functions; Lab 4 discussion	5.7, 5.13	5.3,, 5.6		Lab 4 -Loops and Functions	**Jason Hyde (Plastic Mobile) - User Experience Design**
		14	B	Boolean return value, introduction to Pointers;	8.1, 8.2, 5.9	5.4, 5.6	Tutorial		
		15	C	More pointers, using to receive info back;	5.16	5.7	4		
6	13/Feb	16	A	more on scope; Larger function example - include logic example - Goldbach's conjecture	5.13, 5.16	5.8	Tutorial	NO LAB Friday/Monday of Midterm Week (which is Tuesday Feb 28 @ 1-3pm)	**Dan Di Matteo - Midterm Test Prep**
		17	B	Introduction to Arrays - declaration, initialization, simple use;	6.1, 6.2, 6.3, 6.6, 6.8	6.1, 6.2			
		18	C	Arrays and Pointers; Lab 5 discussion	5.10, 5.12	63	5		
	20/Feb			**READING WEEK, No Classes, No Tutorials, No Labs**					
7	27/Feb	19	A	Midterm Review; solve problems; previous midterms		6.4		Lab 5 - Functions and Arrays and Debugging (DNA Search)	**Alex Rodionov - Engineer's Guide to Debugging Software**
		20	B	Arrays continued, 2D Arrays, Lab 5 Discussion on debugging!	6.7, 6.9, 4.7	6.3	No Tutorial		
		21	C	2D Array Example; Multi-Dimensional Arrays, passing Multi-D arrays;	6.14, 5.10, 5.12	6.3			
8	6/Mar	22	A	Dynamic memory allocation; size of types, sizeof operator, byte addressing, Arrays & dynamic mem; Lab 6 - Game P1	8.1, 8.2 (review) 8.3	6.3, 10.2		Lab 6 - 2D Arrays, Block Diagrams Game Lab Part 1	**Professor Jason Anderson - From Software to Circuits**
		23	B	Pointer Arithmetic; Introduction to strings; Strings - example, I/O,	6.10, 6.11, 6.13	7.2, 7.4	Tutorial		
		24	C	String functions,	6.11, 6.13	7.3, 7.5	6		
9	13/Mar	25	A	More String Functions, 2D Arrays - arrays of strings	6.11, 6.13	8.1, 8.2		**No Lab Graded**; work on Lab 7	**Profs Li/Rose - Strategies for Reversi Game (Lab 7)**
		26	B	Recursion - factorial, printing patterns with recursion	10.1, 10.2, 10.3, 10.9	8.5	Tutorial		
		27	C	More complex Data Structures - structs - Neuron Example; dynamic allocation of structs	7.1, 7.2	10.1	7		

10	20/Mar	28	A	Introduction to Linked Lists	13.1, 13.2	8.10, 10.3		Lab 7 - Game Lab Part 2	**Xavier Snelgrove (Whirlscape) - Machine Learning**
		29	B	Operations on Linked Lists	13.3, 13.4	10.4	Tutorial		
		30	C	More operations on Linked Lists;	13.8, 13.9	10.4	8		
11	27/Mar	31	A	Linked List reprised; Lab 8	13.8, 13.9	9.1, 9.2		Lab 8 - Linked Lists	**Prof. Sven Dickinson - Computer Vision**
		32	B	Searching;	12.1, 12.2, 12.3	9.5, 9.4	Tutorial		
		33	C	Sorting 1 - bubble sort, Selection sort;	12.4, 12.5, 12.6	9.3, 9.7	9		
12	3/Apr	34	A	Sorting 2 - Insertion Sort, QuickSort;	12.7	9.7		Lab 9 - Sorting	**Safdar Mahmood - (Accenture) Telecom Software**
		35	B	Sorting 3 - Quicksort actual code;Lab 9, visualizations: http://www.youtube.com/watch?v=ywWBy6J5gz8 http://www.sorting-algorithms.com	12.7, 12.8	10.4	Tutorial		
		36	C	Binary Trees and Algorithms (Linked List review)	15.1, 15.2	10.7	10		
13	10/Apr	37	A	Binary Trees and Algorithms	15.4, 15.5, 15.6	10.7, 10.8			
		38	B	Binary Trees and Algorithms, continued	15.7, 15.8		No Tutorial		
		39	C	Course Review/Exam Discussion					

Thursday April 13 is last day of Lectures
Friday April 14 is Good Friday, university closed