Course Objectives

Welcome to APS105H, Computer Fundamentals. This course provides an overview of how computers represent and process information, and an introduction to computer programming in C. We will also cover basic searching and sorting algorithms, and explore basic data structures using linked lists. Throughout the course we will emphasize the importance of programming style, and present techniques for writing clear, maintainable code.

By the end of this course, you should have an understanding of how to formulate a problem algorithmically, be able to successfully translate simple algorithms into C, and debug the resulting program.

Contact Information

Instructor	Jonathan Deber
Lectures	M9-10 (BA 1220) W9-10 (BA 1210) F11-12 (BA 1200)
Lab and Tutorial	Lab: T2-4 (SF 1013) Tutorial: F3-5 (~~BA 2175~~ WB 219)
Website	`http://www.ecf.toronto.edu/~deberjon/aps105h/`
Blackboard	`https://portal.utoronto.ca/`
Office Hours	~~TBA~~ M1-2 and W1-2 (BA 2270)
Email	`jdeber -at- cs -dot- toronto -dot- edu`

Resources

The required text for this course is *An Introduction to Computer Science Using C, 2E* by Carter (ISBN: 9781259015267). For more information about this and other recommended books, please see the course website. You are also required to purchase a license for CodeLab ($25 USD), a series of weekly online exercises. For detailed information and instructions, see the course website.

Marking Scheme

Work	**Weight**	**Comment**
CodeLab (∼10)	5%	Each CodeLab is marked out of 2
Assignments (5.5)	20%	A0 (0%), A1 (3%), A2 (3%), A3 (4%), A4 (5%), A5 (5%)
Midterm	25%	Friday March 2, 2012, 3:00 - 5:00 pm
Final exam	50%	During Final Exam Period

Email and Discussion Board

The course has a discussion board on the Blackboard site. This should be your first stop for questions about the course material or general administrative details. Please use descriptive thread names and place each question in a separate thread; this makes it easier for others to find relevant discussions. For example, a single thread entitled “Assignment 2” with several questions in it is less helpful than separate threads with titles like “Use of printf in part 3” or “Expected output for negative numbers”. Also keep in mind that solutions to lab assignments or CodeLabs should never be posted to the discussion board. If you have questions about your code, please come to office hours or ask a TA during a lab.

For individual questions, please contact the instructor via email. I will try to answer all emails by the end of the next day, but it may occasionally take longer, especially on weekends and near due dates. Please include “APS105” in your subject line (e.g., “APS105: Can’t login to Blackboard”) to reduce the likelihood that your message will be misclassified as spam.

You are responsible for all announcements made in lecture and via Blackboard. In addition, please make sure that you are reading (either directly or by forwarding messages to another address) your UTORmail (`@utoronto.ca`) and your ECF (`@ecf.toronto.edu`) email accounts. We (and the rest of the university) may occasionally send important announcements to these addresses.

Term Schedule

M-F Dates	Course Deadlines	Reminders
Jan 16 – Jan 20	A0 due on Saturday Jan 21. CodeLabs begin.	Lectures begin Monday Jan 16.
Jan 23 – Jan 27		Labs and Tutorials begin.
Jan 30 – Feb 3	A1 due on Saturday Feb 4.	
Feb 6 – Feb 10		
Feb 13 – Feb 17	A2 due on Saturday Feb 18.	
Feb 20 – Feb 24		Reading Week, no classes.
Feb 27 – Mar 2	Midterm on Friday Mar 2.	
Mar 5 – Mar 9	A3 due on Saturday Mar 10.	
Mar 12 – Mar 16		
Mar 19 – Mar 23	A4 due on Saturday Mar 24.	
Mar 26 – Mar 30		
Apr 2 – Apr 6	A5 due on Saturday Apr 7.	Good Friday Apr 6, no classes.
Apr 9 – Apr 13		Last day of classes Friday Apr 13.

Tutorials

There are weekly tutorials on Fridays from 3:00 - 5:00 pm in ~~BA 2175~~ WB 219. They begin on January 27th. The TAs will review course material, and will be available to answer your questions about course material or the assignments.

Labs and Lab Assignments

There are 5.5 lab assignments spread over the course of the term. They are intended to provide hands-on experience with the process of programming, since like many skills, programming can only be mastered with practice. Handouts for the assignments will be available on the course website, and will be posted shortly after the previous assignment has been completed. Assignments are generally due by Saturday at 11:59 pm, unless indicated otherwise. Late submissions will not be accepted, barring exceptional circumstances.

Assignments will be auto-marked for correctness, and must function correctly on ECF. This last point may be an issue if you are working on your own computer; if you are, make sure you test your assignments thoroughly on ECF prior to submission. (Note that if you are connected to ECF from home via `ssh`, this is functionally the same as working in the lab.) Assignments will also be marked for style (i.e., a TA will read your code and assign a grade based on how well written it is). Note that it is possible to get high marks for correctness but low marks for style, or vice versa. Submissions will take place via MarkUs (an online submission and grading system), as described on the course website.

All marks will be available via Blackboard. For assignments that have been marked for style, your style marks and comments will be available via MarkUs.

Each lab assignment has practical sessions (Tuesdays from 2:00 - 4:00 pm in SF 1013) where TAs will be available to answer questions about the assignment. These begin on January 24th. You may of course work on the assignments outside of these hours, but we would encourage you to take advantage of the help that is available during those sessions.

The "half" assignment is Assignment 0, which is due at the end of the first week. It is very small, and does not count towards your final grade. It is intended to refresh your memory regarding the edit-compile-submit process, and should not take more than a few minutes to complete. Even though it is worth 0%, we strongly recommend doing it. This is an opportunity to sort out technical problems (e.g., discovering that your ECF account stopped working over the break), and a chance to try out MarkUs.

Assignments are individual work, so you may not collaborate with anyone on them. For more information, please see the section "Academic Integrity".

CodeLab

CodeLabs are a set of online exercises provided by a company called Turing's Craft, and are done on your own time. They are based on a philosophy taken from music education, where students practice basic skills (e.g., scales) to ensure that they have a solid foundation to master more complicated material. We will have CodeLabs most weeks, although some of the later weeks in the term may not have one. A final schedule will be announced later in the term. You must purchase a CodeLab license ($25 USD); for detailed instructions, please see the course website.

Each week's CodeLab is marked out of two. To earn one mark, you must attempt all the exercises and get at least 75% of them correct. For example, if there are 10 exercises, then you must attempt all 10. 75% of 10 is 7.5, so you must get at least 8/10 correct. To earn the second mark, you must get all the exercises correct. You may attempt questions as many times as you want, with some restrictions (see the course website for details). Your total CodeLab mark across all weeks will be weighted as 5% of your term mark. This means that each CodeLab will be worth approximately 0.5% of your course grade, although this percentage may vary slightly depending on the total number of CodeLabs.

Midterm and Final Exam

The course has a midterm during the tutorial time slot on Friday March 2 from 3:00 - 5:00 pm, and a 2.5 hour final exam during the final exam period in April. The midterm is worth 25% of your mark, and the final exam is worth 50%. Both will cover material from lectures, tutorials, the assignments, and indicated chapters in the textbook. The final exam is cumulative.

Missed Work and Re-marks

In the event of an emergency that causes you to miss the midterm or another deadline, please contact the instructor as soon as possible (ideally within 24 hours of the due date). However, it is far easier to find a solution well before the due date, so when possible, please inform the instructor as soon as you know accommodation may be required. In case of illness, have your doctor complete an official U of T medical certificate. For other emergencies, please be prepared to provide other documentation requested by the instructor.

If a piece of work has been mis-marked (including addition errors), you may request a re-mark. For a re-mark to succeed, you must clearly and concisely express what you believe was mis-marked or unfairly marked. To request a re-mark, fill out the form available on the course website, and give it to the instructor within two weeks of the work being returned. For assignment re-mark requests, you do not need to include a copy of your code; for the midterm, please include the test paper. Please note that a re-mark may involve the entire work being reevaluated; as such, your mark may be adjusted up or down, depending on the results of the reevaluation.

Academic Integrity

All of the work you submit must be yours and yours alone. Sharing work with others or using outside resources (e.g., code samples from online sources) without the appropriate citation is considered academic fraud, and is taken very seriously by the University and by this instructor. The consequences can be very severe, and can include penalties such as failing the course, a note on your transcript, or in some cases even expulsion.

So what is considered sharing work? We do not expect you to learn in isolation, and in fact we believe that study groups can be an extremely effective way to learn material. For general studying (e.g., preparing for a test), you can collaborate as much as you want, and can share any resources you have (e.g., lecture notes). However, for assignments, you are allowed to have discussions with other students, but you are not allowed to take written (or electronic) notes away with you. This means that you can work together to decide at a very high level how to structure an assignment (e.g., first we read the input data, then look for a certain string, etc.), but you cannot sit at a computer together and produce the resulting code. You may discuss your code with the instructor or the TAs, but you may not post code to the discussion board (or anywhere else, electronic or otherwise).

It is also worth noting that University regulations dictate that all parties involved in a cheating incident (i.e., both the person who received someone else's work and the person who gave it) are considered equally guilty.

In short, please don't cheat. It's unpleasant for everyone involved, including us, and it simply isn't worth it given the cost-benefit analysis: compare the option of doing poorly by yourself on a 5% assignment versus the option of cheating and potentially failing the course.