
COMPSCI 1XC3

Computer Science Practice and Experience: Development Basics

Pedram Pasandide

McMaster University

Spring/Summer 2024

1 Contact Information

Instructor

Pedram Pasandide pasandip@mcmaster.ca

My Office Hours

Monday 11:30AM - 12:30PM

Wednesday 11:30AM - 12:30PM

If you need, we can always set up appointments, online via Teams.

Teaching Assistants

Grace Croome croomeg@mcmaster.ca

Samkith K Jain kishors@mcmaster.ca

2 Schedule

The course is scheduled from Jun 24, 2024 to Aug 9, 2024.

2.1 Lectures

Attending lectures are not mandatory, but from what I've seen, students who skip out often end up missing some points. We'll discuss assignment instructions right after they're released, which can be really important.

Days	Room	Time
Monday	PGCLL B131	9:30AM - 11:30AM
Wednesday	PGCLL B131	9:30AM - 11:30AM

2.2 Labs

During the lab sessions, we will cover various topics, including Linux installation (not necessary for MacOS users), configuring Visual Studio Code as an Integrated Development Environment (IDE), creating documentation for reports and results, using GitHub, and solving assignments and project-related examples. There will be **pop quizzes during labs** which will be discussed later.

Days	Room	Time
Monday	BSB 244	12:30PM - 4:30PM
Monday	BSB 249	12:30PM - 4:30PM
Wednesday	BSB 244	12:30PM - 4:30PM
Wednesday	BSB 249	12:30PM - 4:30PM

3 Course Description

The main priority of this course is preserving your mental health! You are going to have lots of errors, even when everything looks fine. Nothing is worse than seeing the red colour of an error for a programmer compiling the code. But this is the only way you can learn, by **debugging**. Don't be shy to ask for help.

I'll keep updating the lecture notes on my [GitHub](#). This course provides an introduction to programming, followed by a brief review of working in the Linux (or any Unix based environment) terminal. In the "Fundamentals of C Language" module, students will cover topics such as compiling and executing programs, variable types, and statement structures like loops and if conditions.

The "Intermediate Topics in C" module delves into subjects such as debugging, pointers, and dynamic memory allocation. Moving on to the "Advanced Topics in C" module, students will explore concepts including structs and typedefs, reading a file on C, and write or output the result on a file.

ChatGPT covers the majority of topics addressed in this course, and it is likely to handle a significant portion of the tasks performed by programmers in the near future, as well as creating many new jobs. **If** we ended up having some extra time, I will cover **more advanced topics** in the course. **However, these topics will be optional for students to explore.** One of these optional topics is **Parallel Computing**, which aims to enhance computation speed by utilizing multiple CPUs. Additionally, there will be a brief overview of **Artificial Neural Networks**

(ANN) and a demonstration of solving a simple example. You can pick between these topics and tell me which one you are interested in. It is important to note that these topics are optional.

3.1 Prerequisites and Antirequisite

Prerequisites:

One of

- COMPSCI 1MD3, or
- ENGINEER 1D04.

Antirequisite:

- COMPENG 2SH4
- COMPSCI 1XA3, 2XA3, 2S03
- SFWRENG 2MP3, 2S03, 2XA3

3.2 Resources

I have installed [Linux](#) Ubuntu distribution. Students need only read the lecture notes that I'll update every week. If you are interested in more detailed programming in C you may take a look at [Prof. Barak Shoshany](#), McMaster University, Lecture Notes for Foundations of Modern Scientific Programming.

4 Grading Policy

Quizzes are scheduled to take place during the lab sessions. You will receive an advance notice via Avenue a few days before each quiz, informing you of the specific quiz date. These quizzes will be conducted **in person** and will cover the **basic** topics discussed during our lectures. At the beginning of the lab session scheduled with a quiz, the questions will be released, and you have time to finish it before the lab session is over. **No need to submit anything for these quizzes**, but you **must** show your codes and results to your TAs before you leave the class. TAs may ask you questions and request you to work with your codes, e.g. asking you to run and compile, add/remove, and debug your program. TAs will provide you with feedback and your grade right away, but the grades will be released on Avenue a few days later. During the quizzes using AI

generative models, like ChatGPT, is **NOT** allowed, as I believe quizzes are easy enough. But you can search on the internet, take a look at the lecture notes, talk to your friends, or even having your TA's help. Remember, you are required to participate in **only one** of the lab sessions to take the quiz.

Beside the assignments and quizzes, we will have a final project which is **optional** to do. The topic of the final project will be either about ANN or Parallel Computing with 5-10% bonus. You can pick your own topic as well, but you'll need my approval for the topic you want to work on before starting the project. Here you have an opportunity to develop your ideas, and I'll help you as much as I can.

Option 1: The weights for the final grade without a final project in this course is shown in the following table.

Activity	Score	Weight	Release Week
Assignment 1	5	$5/70 \times 100\%$	2
Assignment 2	10	$10/70 \times 100\%$	3
Assignment 3	15	$15/70 \times 100\%$	4
Assignment 4	20	$20/70 \times 100\%$	5-6
3-4 Quizzes during lab sessions	20	$20/70 \times 100\%$	July 1 - August 7
Total	70	100%	

Option 2: Students who want to do a final project will have the following marking scheme:

Activity	Score	Weight	Release Week
Assignment 1	5	5%	2
Assignment 2	10	10%	3
Assignment 3	15	15%	4
Assignment 4	20	20%	5-6
3-4 Quizzes during lab sessions	20	20%	July 1 - August 7
Final Project	30	30%	7
Total	100	100%	

The final project would be suitable for two groups of students:

- Students who did not receive enough points in other course works, and they would have a chance to redistribute their grade to a final project and possibly having some bonus points. From what I've seen, don't count too much on it since it will be a difficult project.
- Students who have time to explore and learn more about advanced topics, and having feedback about their codes. Some students have done great projects so far.

If some students chose to do the final project (**Option 2**) and their final grade could be higher by **Option 1**, then what we do is:

-
1. Including the final project according to the marking scheme "with" final project (**Option 2**).
 2. Using the marking scheme "without" a final project (**Option 1**).

And then we just use the better of the two grades! In this case you won't need to be worried if you didn't get enough score in the final project.

For the final project, the instructor will select **two topics**, and students can choose one of them. The final project can be completed either **individually** or in **a group of two students**. The topics will be about Parallel computing or ANN, for those students who want to learn more about such topics. As mentioned before, you can pick your topic as well but you need to discuss the topic with me before you start working on it.

Submissions for assignments and projects must be **original** and written by the students themselves from scratch. If students submit code that has been written by someone else, they will receive a **zero grade** for **academic dishonesty**.

While students are allowed to use **ChatGPT** in their assignments only or any [generative AI](#), they **must** acknowledge that they have used it. Additionally, they are required to **understand the meaning** of each line of code provided by ChatGPT. There might be a request for an in-person or online presentation from students to explain the code. It is allowed to include a **short code from other sources**, provided the following **conditions** are met:

1. The copied code is **short** and **not the main component** of the project.
2. The **reference** from which the code was copied is included.

The assignments and the final project code **must be written in C** and **should not contain any errors**. An error means I cannot compile and run the code, which technically means a zero grade.

IMPORTANT! For each assignment or quizzes, I will ask some students randomly or/and based on similarity of submissions to present their codes to me or TAs, by:

- Asking them to compile and run their codes.
- Altering the code like introducing errors or removing a line and asking them to debug and fix the code, using the methods discussed in lectures.
- Applying any topic covered in the lecture notes on their codes.

Ensuring there's no plagiarism in our class is crucial to me. This presentation serves several purposes. It helps verify authorship and ensures fairness in grading. In this course, there are no midterms or final exams. I firmly believe that in practical programming courses, traditional exams hold little value. Verifying your understanding and authorship of the code is vital. Failure to present your code (**for any reason!**) may result in **a loss of up to 100% of the points** for the course work you have been asked to present. This ensures fairness and upholds the integrity of your work.

Do not copy each other's codes. I prefer not to handle plagiarism by involving administrative offices, but I have absolutely no tolerance for copied codes. If I notice code similarity between students, it will be addressed seriously.

4.1 Submission Policy

Late submissions would not be accepted. [MSAF](#) must be submitted online.

We do NOT have any re-weighting option for the missed course work. If you send me MSAF for a quiz, you must contact both me and your TA to take the quiz as soon as possible.

Do **NOT** submit zip files. Submissions in zip file will **not** be marked. PLEASE, follow the instructions of each assignment carefully. You will be asked to submit multiple files for each assignment. Make sure you are submitting all the requested files. There will be no extension for the missed files, and you may even receive a zero grade if I cannot compile and run your code without the missed file.

5 Academic Integrity

We expect you to demonstrate honesty and ethical behavior throughout your learning journey. Your academic achievements should be built upon the solid foundation of integrity and honesty.

It is essential for you to fully comprehend what actions constitute academic dishonesty. Academic dishonesty refers to knowingly engaging in behaviors that lead to unearned academic credit or unfair advantage. Such actions can have severe repercussions, such as receiving a grade of zero on an assignment, losing credit with a notation on your transcript indicating "Grade of F assigned for academic dishonesty," and even facing suspension or expulsion from the university. To familiarize yourself with the various forms of academic dishonesty, please consult the [Academic Integrity Policy](#). Here, we will highlight three examples of academic dishonesty, though this list is not exhaustive:

1. Plagiarism: This occurs when you submit work that is not your own or that has been obtained from another source without proper credit.
2. Improper collaboration in group work: Collaboration is encouraged, but it is essential to respect the boundaries of appropriate teamwork and avoid crossing into dishonest practices.
3. Copying or using unauthorized aids in tests and examinations: Unauthorized aids, such as cheat sheets or devices, should not be utilized during tests or exams.

Remember, fostering a culture of academic integrity is crucial, and we trust that you will make every effort to uphold these principles throughout your academic journey.

5.1 Plagiarism Detection

In certain courses, there might be the utilization of a web-based service called Turnitin.com, which plays a crucial role in identifying the authenticity and ownership of student-submitted work. Additionally, we may employ Moss, a tool that automatically detects similarities in programming code. Students are expected to electronically submit their work either directly to Turnitin.com or through an online learning platform (such as Avenue to Learn) that supports plagiarism detection. This ensures that the submitted work is thoroughly checked for any signs of academic dishonesty.

For more comprehensive information on McMaster University's usage of Turnitin and its policies, please refer to the Turnitin.com website. In conclusion, it is essential to emphasize that plagiarism has no room for escape. We take academic integrity seriously and are committed to upholding its standards.

6 Student Well-being and Code of Conduct

As a strong advocate of student well-being, I deeply resonate with the following quote from my one of my favourite Persian poets:

Human beings are members of a whole, in creation of one essence and soul. If one member is afflicted with pain, other members uneasy will remain. If you have no sympathy for human pain, the name of human you cannot retain. Saadi Shirazi (1210-1292 AD)

Please show respect and understanding for your fellow classmates and TAs. “The Faculty of Engineering is concerned with ensuring an environment that is free of all discrimination. If there is a problem, individuals are reminded that they should contact the Department Chair, the Sexual Harassment Officer or the Human Rights Consultant, as the problem occurs.”