Université d'Ottawa
Faculté de génie

École d'ingénierie et de
technologie de l'information

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Information
Technology and Engineering

Course Syllabus

GNG1106 - Fundamentals of Engineering Computation Fall 2025

Course Description

<catalogue.uottawa.ca> Introduction to computer systems. Problem solving for engineering case studies. Emphasis is on the design of algorithms and their implementation for solving engineering problems using C.

Course Component: Laboratory, Lecture.

The Key to Success

Learning to program is like learning to play an instrument; it takes practice. You need to work through the examples and exercises in this course. Understanding a concept is different from knowing how to use it in your code.

You'll learn a lot by doing the exercises, but the real growth comes when you debug your code. Figuring out why your code isn't working can be frustrating, especially if you're new to programming. You might question if you're smart enough to solve the problem, but remember that even experienced developers find programming challenging. Debugging is an important part of learning and improving your skills.

Textbook, Laptop, Software and Platforms

- ***Textbook***: There is no official textbook. The students are encouraged to consult free online resources about C programming.
- ***Laptop***: each student must have your own laptop and bring it to the lecture.
- ***Exercises***: You will have In-Class and lab Exercises (see Grading),in addition to Assignments.
- ***Brightspace***: will be used as the main teaching platform.
- ***IDE***: see Section "Programming Environment" later.

Class Environment

Lectures	• Lectures run twice a week, each 80 minutes. • Attending lectures is mandatory. • you will have a short "in-class exercises" (ICE). Such exercises are used in part for encouraging students' participation and in part for evaluating students' understanding (for more information, see Grading). • **Students must bring their laptops to the class, to submit their ICEs.**
Labs	• Lab sessions run weekly, **starting from the week of Sept 9**. • There will be no lab during reading week, for week of Oct 28. • Each lab session will be hosted by a TA. In each lab, the students must do. o complete a pre-lab exercise and submit it before the lab (except Lab 1) o complete the required tasks in the lab and have it checked by the TA o Submit it the lab solution by the end of the lab. • The TA may ask additional questions to verify the students' understanding.

Course Objectives

By the end of the course, students will be able to:

- describe the compile, edit, debug, execute cycle.
- recognize and use basic data types, variables, constants, operators, and expressions.
- identify and describe the properties of variables including values, scope, and size.
- Describe the structure of variables and pointers in computer memory.
- choose appropriate language constructs for a given programming task.
- implement, test, and debug algorithms for solving simple problems.
- analyze the behavior of simple programs involving fundamental programming constructs.
- design programs that use compound data types, including arrays, structures, and strings.
- apply dynamic memory allocation and dynamic data structures.

Course grading

The class grade will be based on the following components:

1. **In-class exercises (ICEs)** - **5%.** Each ICE takes at the end of lecture time.
2. **Lab exercises/** - **20%.** There will be 10 sessions.
3. **Midterm- 25%.**
4. **Final Exam** - **50%.** You have to pass the final exam to pass the course

Note: * If the student's final mark is less than 50, his final mark is the mark he gets for the course, namely all other components are erased.

Policies on Late/Missed Submissions and Special Accommodations

- Except for the midterm and final exams, all submissions are all electronically via Brightspace
- Missed submission or absence from an evaluation: Since Fall 2023, the University has started a new policy (Regulation A-8.6. Justification of absence from an examination or of late submission of assignments) concerning this. In this course, we treat late submission the same as absence of evaluation. That is, if you have missed the deadline for the submission of a gradable item, you cannot re-submit it. **If the absence is justified, then the weight of the missed item is automatically moved to the weight of the final exam. There is an exception to this rule, that is, if the total weight of missed non-exam components (i.e., labs, ICE, assignments) reaches or exceeds 50% of the total weight of all non-exam components, whether they are justified or not, the student will be given a grade of INC (``Incomplete") for the course**. In fact, a student having missed this many components should withdraw from the course. To justify an absence or a missed gradable item, the student must perform Declaration of Absence from an EvaluaCon online. The first declaration of absence from any course is approved automatically without supporting evidence (but you still need to go to web portal to declare). Starting from the second declaration, a valid justification is required. When a declaration is submitted, the professor will be able to see and approve/disapprove it using the faculty portal.
- Special accommodations are be given to students registered in Student Academic Success Services (SASS) on a case-by-case basis.

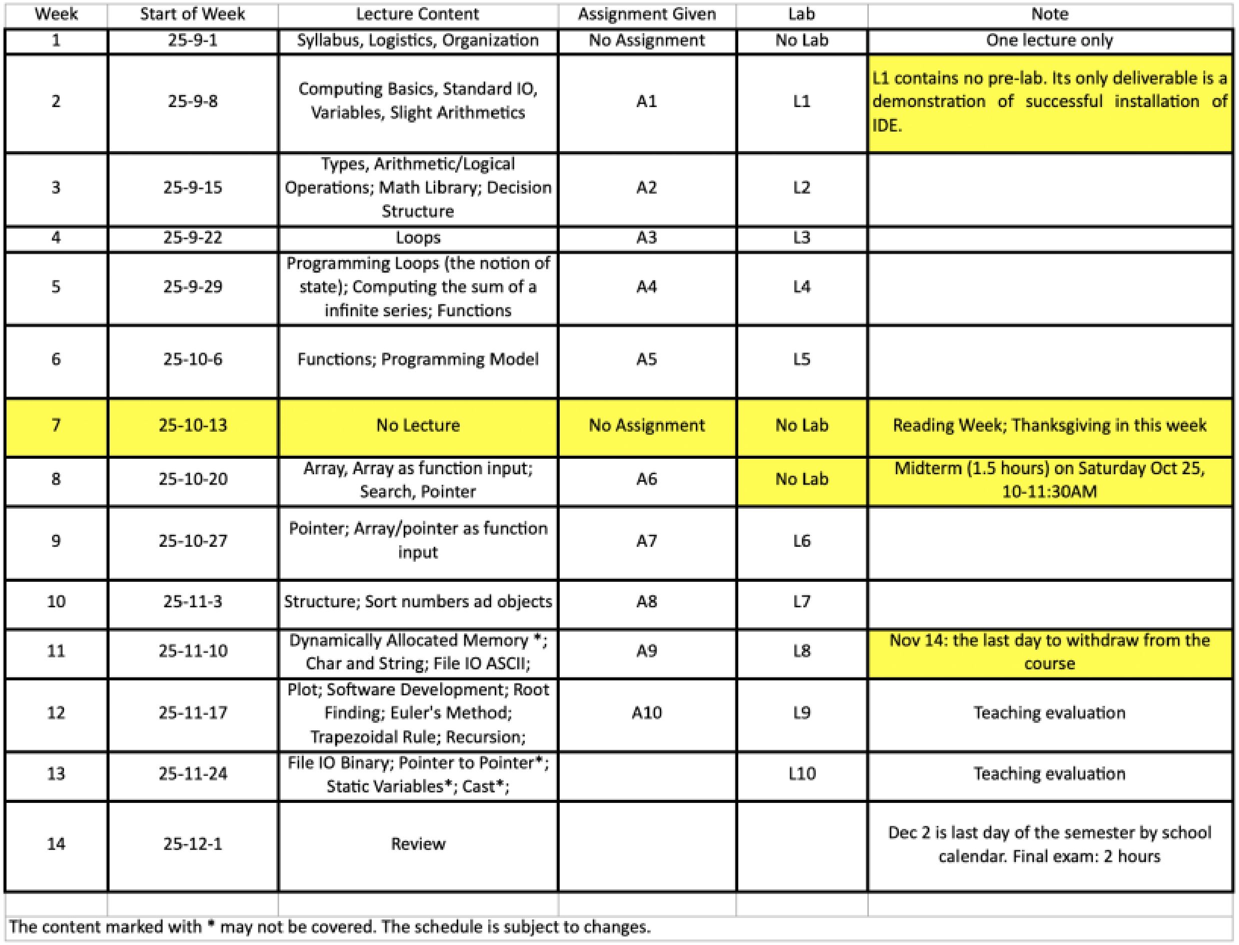
Programming Environment

- A programming environment (referred to as IDE) including a C/C++ compiler MUST be installed on your laptop. The recommended IDE is Code::Blocks, where you can download and install for free from https://www.codeblocks.org/ In rare cases (depending on the laptop you are using), you may need to install a different compiler.
- By Sept 9, all students must have your programming environment (IDE) set up properly. Consult TAs in the lab for assistance if you have difficulties.

Plagiarism and Academic Fraud

Plagiarism (copying and handing in for credit someone else's work) and other forms of academic fraud are serious academic offences that will not be tolerated. Note that the person providing solutions to be copied is also committing an offence as they are an active participant in the plagiarism. The person copying and the person copied from will be reprimanded equally according to the regulations set by the University of Ottawa. **The use of ChatGPT or similar AI tools in any submission is considered an academic fraud.**

Appendix: Weekly Schedule



Week	Start of Week	Lecture Content	Assignment Given	Lab	Note
1	25-9-1	Syllabus, Logistics, Organization	No Assignment	No Lab	One lecture only
2	25-9-8	Computing Basics, Standard IO, Variables, Slight Arithmetics	A1	L1	L1 contains no pre-lab. Its only deliverable is a demonstration of successful installation of IDE.
3	25-9-15	Types, Arithmetic/Logical Operations; Math Library; Decision Structure	A2	L2	
4	25-9-22	Loops	A3	L3	
5	25-9-29	Programming Loops (the notion of state); Computing the sum of a infinite series; Functions	A4	L4	
6	25-10-6	Functions; Programming Model	A5	L5	
7	25-10-13	No Lecture	No Assignment	No Lab	Reading Week; Thanksgiving in this week
8	25-10-20	Array, Array as function input; Search, Pointer	A6	No Lab	Midterm (1.5 hours) on Saturday Oct 25, 10-11:30AM
9	25-10-27	Pointer; Array/pointer as function input	A7	L6	
10	25-11-3	Structure; Sort numbers ad objects	A8	L7	
11	25-11-10	Dynamically Allocated Memory *; Char and String; File IO ASCII;	A9	L8	Nov 14: the last day to withdraw from the course
12	25-11-17	Plot; Software Development; Root Finding; Euler's Method; Trapezoidal Rule; Recursion;	A10	L9	Teaching evaluation
13	25-11-24	File IO Binary; Pointer to Pointer*; Static Variables*; Cast*;		L10	Teaching evaluation
14	25-12-1	Review			Dec 2 is last day of the semester by school calendar. Final exam: 2 hours

The content marked with * may not be covered. The schedule is subject to changes.