Université d'Ottawa
Faculté de génie

École de science informatique
et de génie électrique

uOttawa

L'Université canadienne
Canada's university

University of Ottawa
Faculty of Engineering

School of Electrical Engineering
and Computer Science

GNG1106 D
Fundamentals of Engineering Computation
Winter 2020

Instructor: Dr. Mohamed Hossam Ahmed **E-mail:** mahme3@uottawa.ca
Office: SITE 5026D
Office Hours: Check course website on BrightSpace

Teaching Assistants: Ayse Rumeysa Mohammed **E-mail:** amus037@uottawa.ca
Alaa Eddin Alchalabi **E-mail:** aalch040@uottawa.ca
Masoumeh Soflaei Shahrbabak **E-mail:** msofl083@uottawa.ca

Weekly Schedule:

LEC 1	Tuesday 10:00 – 11:30	STE A0150
LEC 2	Thursday 8:30 – 10:00	STE A0150
Lab	Check course schedule	

- All components of the course (i.e laboratory reports, assignments, etc.) must be fulfilled otherwise students may receive an INC as a final mark (equivalent to an F).

Description:
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 Objectives:
At the end of this course, the student shall be capable of:
- Solving real engineering problems using computer programs developed with a high level language,
- Applying basic software engineering principles for creating these computer programs,
- Programming with the C language using the following programming concepts: control structures, functions, arrays, character strings, searching and sorting, files, structures and recursion.
- See the course Schedule in the course Web Site for a detailed list and schedule of the course topics.
- General learning outcomes can be found at the end of this syllabus.

Course Website: https://uottawa.brightspace.com/d2l/le/content/139737/Home

- You will use it for
  - checking announcements/schedules/other admin information
  - accessing course materials
  - submitting assignments/projects
  - interacting with peers/instructor/TA

Textbook & Supplies:

- There is no official textbook for the course. The following are suggested books.
  - "***Engineering Problem Solving with C***", Delores M. Etter, Pearson/Prentice Hall, Upper Saddle River, New Jersey.

- "***GNG 1106 Laboratory Manual***", G. Arbez and M. Eid, available from the course website.

Laboratory Sessions:

- There will be programming laboratory sessions given in the laboratory to allow you to explore techniques and concepts to assist you in your learning of the course material. These lab sessions are there to help you.
- Lab exercises will be posted on the course Web site ahead of time. So do review the exercises before coming to the sessions.
- Read the GNG Lab manual before your first scheduled lab session.
- Lab exercises are evaluated and marked.

Assignments:

- For the first half of the course, weekly assignments must be complete and submitted for marking via BrightSpace. These assignments will get you to practice the concepts learned in class and the lab sessions.
- The assignments are to be submitted to BrightSpace. They shall not be accepted by e-mail.
- The assignments will be marked and returned, also via the BrightSpace.
- Late assignments will not be accepted and hence will be assigned a mark of zero.

Project:

- For the second half of the course, you shall be working in teams of 2 to complete and Engineering Computation project. Weekly deliverables are handed in for the projects which will follow a 5-step problem solving methodology. This project will allow you to apply the basics learned in the first part of the course as well as more advanced strategies to be introduced in the latter part of the course.
- The project deliverables are to be submitted to BrightSpace. They shall not be accepted by e-mail.
- The project deliverables will be marked and returned, also via the BrightSpace.
- Late assignments will not be accepted and hence will be assigned a mark of zero.

Marking:

Course Grade Requirements	Marking Scheme
• Assignments	**10%**
• Projects	**10%**
• Labs	**10%**
• Midterm Examination	**20%** (scheduled on Tue Mar 3rd class time)
• Final Examination	**50%**

- A missing assignment/lab/project/midterm exam will only be excused if it is due to severe illness or other events with extreme measure, in which case a certificate must be presented.
- The weight of the excused missing component will be shifted to the final exam.
- Assignments are posted on the BrightSpace. Solutions are to be submitted online through BrightSpace ONLY. Late assignment submission (with penalty) will be available through BrightSpace for 24 hours after the due time. Late assignments will have a penalty of 2% per hour (e.g., if the assignment is submitted within the fifth hour after the due date, it will have a 10% penalty). Assignment submission by email is not allowed.

Plagiarism:

Plagiarism (copying and handing in for credit someone else's work) is a serious academic offence 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. Please refer to the section on academic offences in the Undergraduate Calendar and the notice on plagiarism on the University of Ottawa website for additional information: www.uottawa.ca/academic/info/regist/crs/0305/home_5_ENG.htm.

Learning Outcomes:

Learning Philosophy:

- Develop conceptual and problem solving capabilities to apply engineering computation to solving engineering problems; this requires a capability of abstraction and creativity.
- Software development requires logical thinking and management of many details; the course allows students to develop their logical thinking to solve problems with computing logic.
- Study and discuss the principles of software development, including the understanding of the logic of the computer and the principles of software engineering.

Learning in this course means that you shall develop approaches to analyze a problem and solve it with computing software. To achieve this, the teaching approach respects the following principles:

- Introduction to programming: Understand the logic of software by developing conceptual models of the logic of a computer language and thus the logic of software.
- Computing strategies for solving problems: Study various computing strategies to solve engineering problems. The first strategies are simple such as computing calculation to evolve to more complex strategies such as numerical methods.
- Five step problem solving method: During the 2nd half of the course, solve engineering problems by following a 5 step problem solving method. This method will be applied in a computing project.