# Programming Fundamentals for Software and Computer
# Lab 1 – Fall 2021

Department of Electrical & Computer Engineering
**University of Calgary**

*Written by: M. Moussavi, PhD, PEng*

## Instructors:

- Dr. Mahmood Moussavi (B01, B02)
- Dr. Maan Khedr (B03, B04)

## Objectives:

This is much smaller lab assignment than most of the future labs. It contains several simple exercises, that are designed to help you to:

- Get familiar with the C program development environment, compiling running a C program
- Review some of the very basic programming features in C.

## Please heed this advice:

.

- When writing code (C or C++) make sure the following information appears at the top of your code:
  - File Name
  - Assignment and exercise number
  - Lab section
  - Your name
  - Submission Date:

  **Here is an example:**

```
/*
 *   File Name: lab1exe_B.c
 *  Assignment: Lab 1 Exercise B
 *  Lab section: Your Lab Section (B01, B02, B03, or B04)
 *  Completed by: Your Name (plus your partner(s) name for group exercises)
 *  Submission Date: Sept 18, 2021
 */
```

- Some exercises in this lab and future labs will not be marked. Please do not skip them, because these exercises are as important as the others in learning the course material.
- In courses like ENSF 337 some students skip directly to the exercises that involve writing code, skipping the sections such as "**Read This First**", or postponing the diagram-drawing until later. That's a bad idea for several reasons:
  - "**Read This First**" sections normally explain some of technical or syntax details that may help you to solve the problem or may provide you with some hints.
  - Drawing diagrams is an important part of learning how to visualize memory used in C and C++ programs. If you do diagram-drawing exercises at the last minute, you won't learn the material very well. If you do the diagrams first, you may find it easier to understand the code-writing exercises, so you may be able to finish them more quickly.

## Due Dates:

Due date for this lab assignment is: **Thursday Sept 23, before (2:00 PM)**

All of your work should be submitted in a single file in PDF format. For instructions about how to provide your lab reports, study the posted document on the D2L called: *How to hand in your lab assignment.*

## Important Notes:

1. Ask for help but don't copy.
   - You can get help from lab instructor(s), TAs, or even your classmate as long as you do not copy other people's work
   - If we realize that even a small portion of your work is a copy from a classmate both parties (the donor, and the receiver of the work) will be subject to academic misconduct (**plagiarism**). More importantly if you give exercise solutions to a friend you are not doing him/her a favor, as he/she/them will never learn that topic and will pay off for this mistake during the exam or quizzes. So, please do not put yourself and your friend into the position of academic misconduct.
2. Some lab exercises may ask you to draw a diagram, and most of the students prefer to hand-draw them. In these cases, you need to **scan** your diagram with a scanner or an appropriate device such as your mobile phone and insert the scanned picture of your diagram into your PDF file (the lab report). A possible mobile app to scan your documents is **Microsoft Lens** that you can install it on your mobile device for free. Please make sure your diagram is clear and readable, otherwise you may either lose marks, or it could be impossible for TAs to mark it at all.

## Marking Scheme:

You shouldn't submit anything for the exercises that are not marked.

| Exercise | Marks |
|---|---|
| A | no marks |
| B | 4 marks |
| C | 6 marks |
| D | 5 marks |
| E | 10 marks |

# Exercise A: Creating a C source file, building and running an executable

## Read This First:

There are no marks for this exercise, but if this is your first attempt to develop a C program, please do it so that you start to become comfortable with program development in C and particularly using your favorite development environment such as (Cygwin + Notepad++), Geany, or Xcode. Before starting this exercise, please read the slides on "Getting Started".

## What to Do:

- If you haven't already done so, make a folder called ENSF337 on your computer.
- Within your ENSF337 folder, make another folder called lab1.
- Start up your favorite text editor, and type in all the following C code:

```
#include <stdio.h>

int main(void)
{
  int a = 0, b = 0;
  printf("Please enter a value for variable a:\n");
  scanf("%d", &a);
  printf("Please enter a value for variable b:\n");
  scanf("%d", &b);
  printf("The values of a and b are %d for a and %d for b.\n", a, b);
  printf("The value of a %% b is %d.\n", a % b);
  return 0;
}
```

- Now save your file as: `lab1exe_A.c`
- In Cygwin Terminal (or mac terminal) navigate to your working directory (the same directory that you saved your `lab1exe_A.c)`, then enter the command:

```
gcc -Wall lab1exe_A.c
```

If the command succeeds, an executable file called a.exe will have been created. If the command fails -- which will be indicated by one or more error messages--go back to your editor and fix the code that you have typed incorrectly and try gcc again.

- Once you have an executable, run it a few times by using the command

```
./a.exe
```

over and over. Try different inputs each time; see what happens if you enter letters or punctuation instead of numbers.

**Notes:**

- You don't need to type `./a.exe` over and over! You can use the up-arrow key on your keyboard to retrieve previously entered commands.

- For those who use a Mac computer terminal, the default executable file created automatically is called **a.out** instead of **a.exe.**

### What to Submit:

There is nothing to be submitted for this exercise.

## Exercise B - Variables and Basic Arithmetic (4 marks)

Recall that variables are used to store values in a program. However, before variables can be used, they must first be created (declared/defined). This exercise will get you exposed to variable creation as well as basic arithmetic.

### What to Do:

Copy the following incomplete C program into your favourite text editor or IDE:

```
#include <stdio.h>
int main()
{
   double num1 = -34.5;
   double num2 = 98.7;

   double sum;          // sum of num1 and num2
   double sumSquared;   // the square of num2 plus num2

   // 1) Add the two numbers and store the result in the variable 'sum'

   // 2) Compute the square of the sum and store the result in the variable 'sumSquared'
   //    Use the variable 'sum' (computed above) for this computation


   printf( "The sum squared is: %f \n", sumSquared);

   // 3) Now double the sum squared value and store the result in 'sumSquared'

   printf( "The sum squared is now: %f \n",  sumSquared);
```

```
    return 0;
}
```

This code contains most of a program to perform various numerical calculations. What you need to do is:

- At the point labelled 1), write the code to add the values of **num1** and **num2** and to store the result in the variable **sum**.
- At the point labelled 2), write the code to compute the square of the sum of **num1** and **num2** and to store the result in the variable **sumSquared**. Use the value of the variable **sum** computed in the previous step.
- At the point labelled 3), double the value of the **sumSquared** variable and store the result in **sumSquared**.
- Compile your program.
- Run the program and record the output.

### What to Submit:

In an MS Word document copy and paste your **source code** (.c file) and the **program output (screenshot of terminal output)**. Now save your **lab-report**.

Note: You have to still add next exercise to your lab-report before being ready to create a PDF file and submit it electronically.

## Exercise C. Operator Precedence (6 marks)

In the lectures, you were introduced to operator precedence in C. In this Task, you are asked to predict the value of the following expressions by using the proper operator precedence.
First, assume that:

```
double z = 0;
double x = 2.5;
double y = -1.5;
int m = 18;
int n = 4;
```

Now, what are the values of z in the following expressions? Show your work.

**a)** `z = x + n * y – (x + n) * y;`
**b)** `z = m / n + m % n;`
**c)** `z = n / m + n % m;`
**d)** `z = 5 * x – n / 5;`
**e)** `z = 1 – ( 1 – ( 1 – ( 1 – ( 1 – n))));`
**f)** `z = sqrt(sqrt((double)n);`

### What to Submit:

In your MS Word document created for previous exercise, type your answers. Don't forget to show steps taken towards the final answer

Note: You have to still add next exercise to your lab-report before being ready to create a PDF file and submit it electronically.

## Exercise D: Mathematical Expressions (5 marks)

For this task, write a program to read in an angle in units of radians and compute the sine of the angle. However, in addition to using the built-in sine function, you will also compute the Taylor series approximation, which is given by

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

The pseudo-code for your program is as follows:

- Prompt for input angle in units of radians.
- Read input angle in units of radians.
- Compute and output the sine of the input angle using built-in trigonometric functions.
- Compute and output the Taylor series approximation of $\sin(x)$ including terms up to order seven (i.e., $x^7$).

Write your source code in a file called **lab1_exe_D.c** and compile it into a program called **sine**. Remember, in order to use the built-in $\sin(x)$ or $pow(x, y)$ functions, you must include the **math.h** file. To get you started you can use the following template

```
#include <stdio.h>
#include <math.h>


int main()
{
   // Develop the body of the program here

   return 0;
}
```

Once you have written and compiled your code, test your program by running it and recording the outputs for following input values

a. 0 radians (0 degrees)

b. 0.5 radians (approximately 28.65 degrees)

c. 1.0 radians (approximately 57.30 degrees)

d. 2.5 radians (approximately 143.24 degrees)

### What to Submit:

In you MS Word document created for previous exercise add the copy of your C program, a screenshot of the program output to terminal, and a table of results for the recorded results for inputs a through d.
Note: You have to still add next exercise to your lab-report before being ready to create a PDF file and submit it electronically.

## Exercise E: Problem Solving (10 marks)

### Problem Statement

You have been given the problem of writing an algorithm to solve quadratic equations. The input to the algorithm is the three coefficients, $a$, $b$ and $c$ of the quadratic equation to be solved

$$ax^2 + bx + c = 0$$

Then, the solutions to the quadratic equations are given by

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

The algorithm should be able to output both real and complex results (i.e., those results which require taking the square root of negative number; $i = \sqrt{-1}$). As an example of a complex result, consider the following equation:

$$x^2 + 2x + 1 = 0$$

Which its solutions are:

$$\begin{aligned} x &= \frac{-2 \pm \sqrt{2^2 - 4 \cdot 1 \cdot 2}}{2 \cdot 1} \\ &= \frac{-2}{2} \pm \frac{\sqrt{-4}}{2} = -1 \pm i \\ &= -1 + i \quad \text{and} \quad -1 - i \end{aligned}$$

Note, however, that a computer **cannot** compute the square root of a negative number. If this is required, your algorithm should compute the square root of the absolute value of the number and prefix the output (scaled by the denominator) with an "$i$". Mathematically, this is equivalent to writing the solution to the quadratic equation as

$$x = \frac{-b}{2a} \pm i \cdot \frac{\sqrt{|b^2 - 4ac|}}{2a} \quad \text{for} \quad b^2 - 4ac < 0$$

### What to Do:

Write your source code in a file called **lab1_exe_E.c** and compile it into a program called **quadratic**. Remember, in order to use the built-in math functions, you must include the **math.h** file. To get you started you can use the following template

```
#include <stdio.h>
#include <math.h>
```

```
int main()
{
   // Develop the body of the program here

   return 0;
}
```

### What to Submit:

In your MS Word document created for previous exercises add the copy of your C program, screenshot of the program output for three different coefficients sets, and a table of results to summarize tour test results with three columns (a, b, c) and two columns for roots (root 1, root 2)
Note: Now if you are done with other exercises, you are ready to create a PDF file and submit it electronically.

### Full Submission:

Create a folder named: lab 1 – UCID-Name

- In your folder, add the report pdf file.
- In your folder add subfolders for exercises with code requirement (for example Exercises b, d, and e) and copy your source code files to them

Compress the folder and submit to D2L