Installation

There are two recommended Integrated Development Environment (IDE) for C programming for this course.

1) **Visual Studio (VS) code**
2) **CodeBlocks**

Note: Installing only one of this environment is sufficient.

1) VS Code Installation

VS Code is a lightweight, free, and highly extensible code editor developed by Microsoft in which C/C++ entensions can be added for running C codes. Please carefully follow the instructions below to complete the installation.

Step 1 (Windows): Install GCC Compiler in Windows To Run C Program. Follow THIS tutorial. Alternatively, follow this video

Note: iOS or Linux user should skip this step and GCC is already installed in these operating systems

Step 2: Install VSCode and C/C++ extension and .Run extension in VSCode. Follow THIS tutorial.

Test installation:

1. Open VS Code, open any folder and create a file with the extension .c for the C file.
2. Write your first C code:

```c
/* Print a message on the screen */
#include<stdio.h>

int main()
{
    printf("Welcome to UCL\n");
    return 0;
}
```

3. Run the code directly using the play button on the upper right corner. It will compile and run the code directly.
4. You are now all set to compiling and running your amazing C programming code.

2) CodeBlocks Installation

CodeBlocks is an open-source IDE that is known for its simplicity and ease of use, making it a good choice for beginners.

For its installation and how to run your first code using CodeBlocks, following THIS video tutorial.

Lab Session 1:

Getting familiar with the C programming syntax, programming environment, introduction to variable types and string input output

1. **Basic variable types in C**

```c
char c; // character data type
int i; // integer data type
float f; // floating point data type
```

a. Integer variables (int):
An int is a data type used to store whole numbers (integers). It can represent both positive and negative integers.

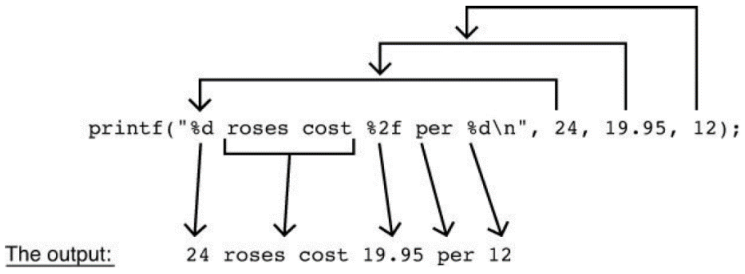
b. Floating-Point Variables (float):
A float is a data type used to store decimal numbers (real numbers). It can represent numbers with fractional parts.

c. Character Variables (char):
A char is a data type used to store a single character, such as a letter, digit, or symbol. It's enclosed in single quotes (e.g., 'A').

2. **Conversion Characters:**
In C, conversion characters are placeholders used in format strings with *printf* and *scanf* functions to specify the data type of the variable being processed. Here are some common conversion characters:

Conversion character	Description
%d	Used for int variables in both printf (for output) and scanf (for input).
%f	Used for float variables in both printf and scanf
%c	Used for char variables in both printf and scanf
%s	Used for string (char array variables) in both printf and scanf

3. **String input and output**



Exercise 1: Use of int and float

The below C program takes user age as input and print the entered age.

Write this code in your C programming editor, compile and run.

```c
#include<stdio.h>
int main()
{

    // Integer variable
    int age = 25;

    // Input age from user
    printf("Enter your age: ");
    scanf("%d", &age);

    // Displaying the age
    printf("You entered: %d\n", age);

    return 0;
}
```

a. What happens if you enter a real number as age input?
b. How to update this code to enable taking real number input?
c. What does '\n' do? From the recommended textbooks, read about the 'escape sequences'.

Exercise 2: Use of float and arithmetic operation

Write a C program that calculates the area of a circle. The program should take the radius of the circle as input from the user (a floating-point number) and then calculate and display the area.

Exercise 3: Use of char, scanf(), getch(), getche() functions

Write a C program that takes your grade as input (A, B, C, D) using scanf(), and displays your grade.

a. Replace scanf() with getch(). What difference do you notice in the output?
b. Now replace getch() with getche(). What difference do you notice in the output?
c. When printing a char type variable, instead of %c , use %d. Does the program returns an error? What do you observe when entering A, then B, then C, and then D in individual runs?

Hint: In addition to stdio.h, include the conio.h header file for using get character functions.

Exercise 4: Use of scanf, gets, fgets, printf, sprint, puts functions

The below C program takes a string type as input and displays the string.

Write this code in your C programming editor, compile and run.

```c
#include <stdio.h>

int main()
{
    // array to store string taken as input
    char color[20];

    // take user input
    printf("Enter your favourite color: ");
    scanf("%s", color);

    // printing the input value
    printf("Your favourite color is: %s.", color);

    return 0;
}
```

a. Why there in no '&' sign before variable color in the scanf statement?
b. Update this code to take two inputs (your name and favourite color) as inputs, and display the sentence to say your name and favourite colour.

   This is how your output should look:

```
Enter your username: Sophia
Enter your favorite color: pink
Hello Sophia, your favorite color is pink
```

c. Instead of 'scanf' for taking input, use 'gets'.
d. Instead of 'printf' to display the entered string, use 'puts'.
e. Can the 'puts' function takes two string variables? Explore the use of 'sprint' and 'puts' for this task.
f. Here is an example how you can use fgets

```c
fgets(color, sizeof(color), stdin);
```

   Change the above code to use fgets. Experiment with defining color[5]. Enter 'yellow' as your favorite color. What do you observe?

Exercise 5: Use of different variable types and input output functions

Write a C program to input and display information about a student. The program should do the following:

a. Ask the user to enter the student's name as a string.
b. Ask the user to enter the student's age as an integer.
c. Ask the user to enter the student's GPA as a floating-point number.
d. Ask the user to enter the student's grade (A, B, C, D, or F) as a character.
e. Display all the information entered, including the student's name, age, GPA, and grade in a single sentence.