

Programming Fundamentals Lab



Lab # 02 (Week 04)

Getting Started with Code

Instructor: Fariba Laiq

Email: fariba.laiq@nu.edu.pk

Course Code: CL1002

Semester Fall 2023

Programming:

Programming is a way to “instruct the computer to perform various tasks”.

Programming Language:

A programming language provides a way for a programmer to express a task so that it can be understood and executed by a computer.

C Introduction:

C is a **high-level and general-purpose computer programming language**. It was created in the 1970s by Dennis Ritchie and remains very widely used and influential. It is mainly used to develop many applications and operating systems such as Windows, Linux and other complicated programs such as the Oracle database, drivers, Python interpreter, and games. It is considered a programming foundation in learning any other programming language. That is why it is also called as the **mother of all programming languages**.

Why learn C?

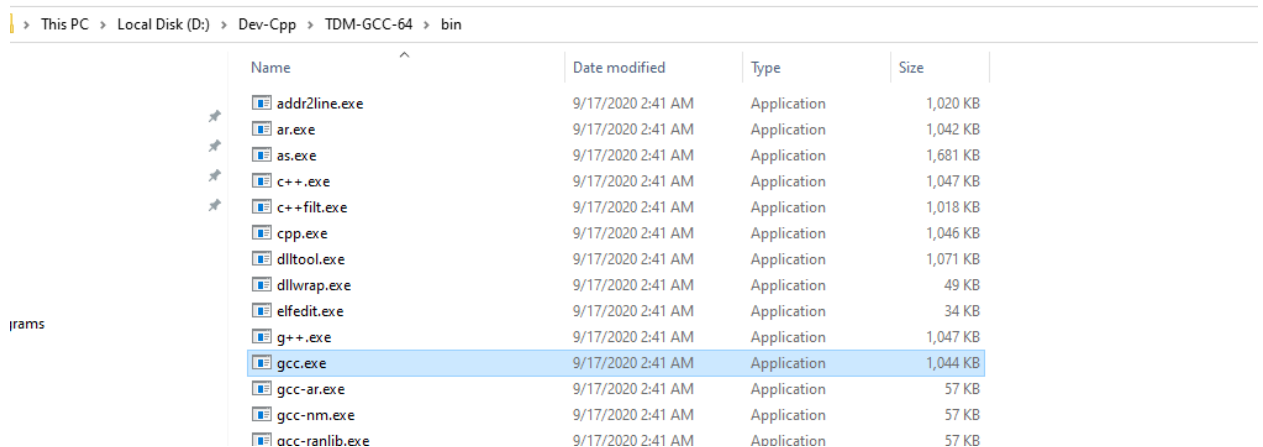
- It is one of the most popular programming languages in the world
- If you know C, you will have no problem learning other popular programming languages such as Java, Python, C++, C#, etc., as the syntax is similar
- C is very fast, compared to other programming languages, like Java and Python

GCC Compiler:

GCC stands for GNU Compiler Collection. It is used to compile programs written in C, C++, Java, ada, and Fortran, etc.

Compile and Run your First C program Using Command Prompt

- If you have already Dev Cpp installed, locate the gcc compiler, in the Dev Cpp folder.

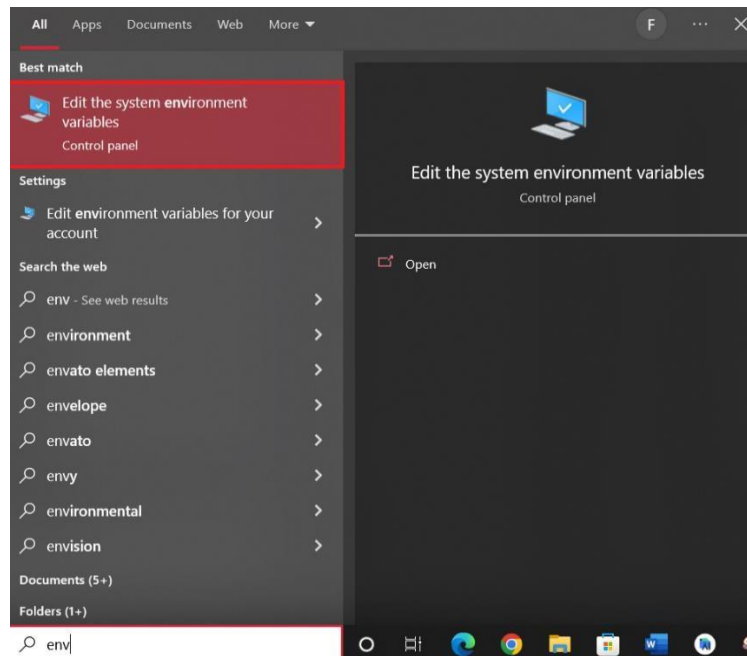


Name	Date modified	Type	Size
addr2line.exe	9/17/2020 2:41 AM	Application	1,020 KB
ar.exe	9/17/2020 2:41 AM	Application	1,042 KB
as.exe	9/17/2020 2:41 AM	Application	1,681 KB
c++.exe	9/17/2020 2:41 AM	Application	1,047 KB
c++filt.exe	9/17/2020 2:41 AM	Application	1,018 KB
cpp.exe	9/17/2020 2:41 AM	Application	1,046 KB
dlltool.exe	9/17/2020 2:41 AM	Application	1,071 KB
dllwrap.exe	9/17/2020 2:41 AM	Application	49 KB
elfedit.exe	9/17/2020 2:41 AM	Application	34 KB
g++.exe	9/17/2020 2:41 AM	Application	1,047 KB
gcc.exe	9/17/2020 2:41 AM	Application	1,044 KB
gcc-ar.exe	9/17/2020 2:41 AM	Application	57 KB
gcc-nm.exe	9/17/2020 2:41 AM	Application	57 KB
gcc-ranlib.exe	9/17/2020 2:41 AM	Application	57 KB

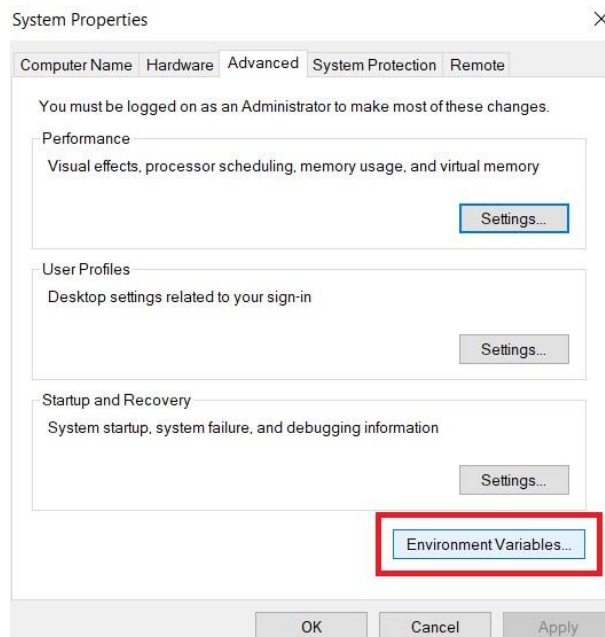
- Copy the path.

- Now we will add the path to the gcc compiler to system environment variables so that we don't have to give the full path to the GCC compiler when we run the compiler from the command prompt. Use the following steps to add the path to gcc to system environment variables:

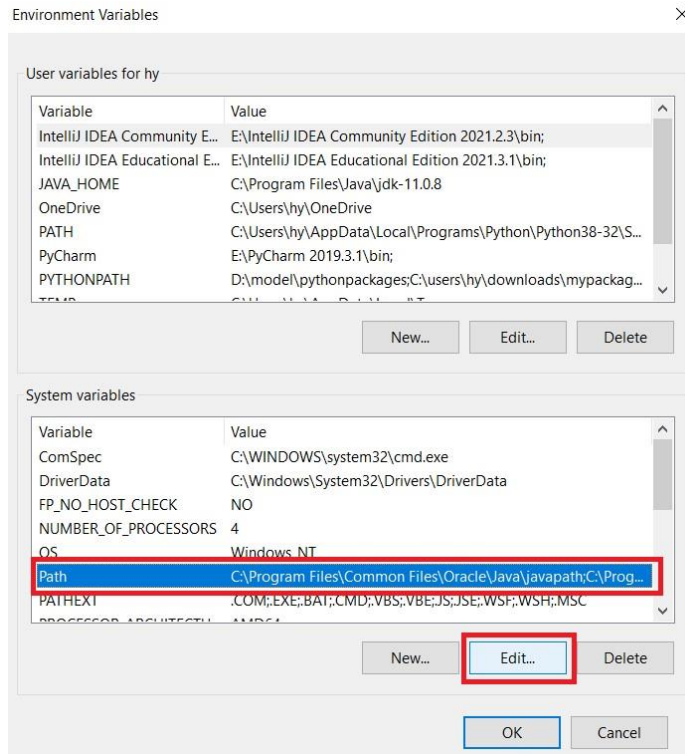
- Type **environment variables** in the search bar of Windows and click on **edit the system environment variables**.



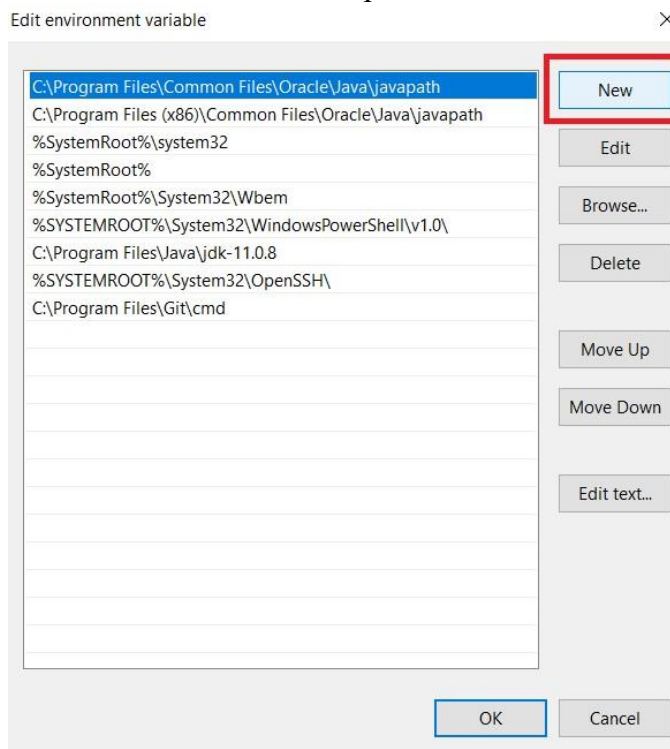
- Click on the **environment variables**.



- From the **system, variables** click on the **Path** and select **Edit**.



-
- Click on **New** to add a new path.



-
- Paste the path to the **bin** folder where your GCC compiler is located.

Now to check if the path to gcc has been successfully set or not, open the command prompt and type gcc --version and hit enter. If the command runs without any error, it means the path has been set successfully.

- Now open notepad and paste the following code:

```
#include <stdio.h>

int main() {
    printf("Hello, World!");
    return 0;
}
```

- Save your file as **program.c** (or anyname followed by the .c).
- Open cmd and go to the directory where your C program is located. In case the program is located on desktop type: **cd desktop**
- To compile your C program into object code, using the GCC compiler type the following command in cmd.

gcc -c source_file_name -o object_file_name

- Where the object file name is the name you wish to name the object file after compiling your program.
- In our case, we will write:

gcc program.c -c prog

- This command will compile your program and create an object file.
- Now to convert the object file into an executable file, use the following command.

- **gcc prog.o -o p**

- Where p is the name of the executable file.
- Now to execute the file simply type the name of the executable file and hit enter.
- **p**

```
C:\Users\hy\Desktop>gcc -c program.c -o prog.o
C:\Users\hy\Desktop>gcc prog.o -o p
C:\Users\hy\Desktop>p
Hello, World!
C:\Users\hy\Desktop>_
```

If you want to create an executable file directly you can use the following command:

gcc program.c

If you don't specify any name for executable file, by default it's name will be a.

```
C:\Users\hy\Desktop>gcc program.c
C:\Users\hy\Desktop>a
Hello, World!
C:\Users\hy\Desktop>_
```

Structure of a Basic C Program

Structure of C Program	
Header	#include <stdio.h>
main()	int main() {
Variable declaration	int a = 10;
Body	printf("%d ", a);
Return	return 0; }

The printf() in C

In C programming language, printf() function is **used to print the (“character, string, float, integer, or any other data type variable”)** onto the output screen. It is defined in stdio.h.

Escape Sequences in C

They are primarily used to put nonprintable characters in character and string literals. For example, you can use escape sequences to put such characters as tab, next line, backslash, etc into an output stream.

Constant	Meaning
\a	Alert
\b	Backspace
\f	Form feed
\n	New line
\r	Carriage Return
\t	Horizontal tab
\v	Vertical tab
\'	Single quote
\"	Double quote
\?	Question mark
\\	Backslash
\0	Null

Escape sequence	Description	Example	Output
\n	New line	<code>printf("Hello \n World");</code>	Hello World
\t	Horizontal tab	<code>printf("Hello \t World");</code>	Hello World
\'	Single quote	<code>printf("Hello \'World\' ");</code>	Hello 'World'
\"	Double quote	<code>printf("Hello \"World\" ");</code>	Hello "World"
\\	Backslash	<code>printf("Hello \\World");</code>	Hello \World

Data Types:

Data type tells the compiler the size of the variable, and the type, of the data it stores.

Data type	Size(bytes)	Range	Format String
char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
short	2	-32,768 to 32,767	%d
unsigned short	2	0 to 65535	%u
int	2	32,768 to 32,767	%d
unsigned int	2	0 to 65535	%u
long	4	-2147483648 to +2147483647	%ld
Unsigned long	4	0 to 4294967295	%lu
float	4	-3.4e-38 to +3.4e-38	%f
double	8	1.7 e-308 to 1.7 e+308	%lf
long double	10	3.4 e-4932 to 1.1 e+4932	%lf

Variable

- A named memory location where data is stored is called variable.
- A quantity whose value may change during execution of the program is called variable. It is represented by a symbol or name.
- **Variable** is name of *reserved area allocated in memory*. In other words, it is a *name of memory location*.
- It is a combination of "vary + able" that means its value can be changed.
- `int data=10 // Here data is variable.`

Rules for defining variables

- A variable can have alphabets, digits, and underscore.
- A variable name can start with the alphabet, and underscore only. It can't start with a digit.
- No whitespace is allowed within the variable name.
- A variable name must not be any reserved word or keyword, e.g. int, float, etc.
- Variable name must be meaningful and should not be too long (convention).

Valid variable names:

`int a;`

`int _ab;`

`int a30;`

Invalid variable names:

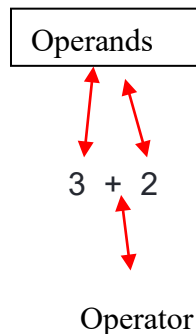
`int 2;`


```
int a b;
```

```
int long;
```

Operators

Operators are special symbols that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.



Here, `+` is the operator that performs addition. `2` and `3` are the operands and `5` is the output of the operation.

```
int sum1 = 100 + 50;           // 150 (100 + 50)
int sum2 = sum1 + 250;         // 400 (150 + 250) int
sum3 = sum2 + sum2;           // 800 (400 + 400)
```

Arithmetic Operators

Arithmetic operators are used to perform mathematical operations like addition, subtraction, multiplication etc.

Operator	Meaning	Example
+	Add two operands	<code>x + y</code>
-	Subtract right operand from the left	<code>x - y</code>
*	Multiply two operands	<code>x * y</code>

/	Divide left operand by the right one.	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)

Assignment operators

Assignment operators are used to assign values to variables.

`int a = 5` is a simple assignment operator that assigns the value 5 on the right to the variable a on the left.

Operator	Example	Equivalent to
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5

Expression

An expression is a combination of operators, constants and variables.

Examples:

x+y

x+y-1

x+1

Precedence of Operators

There can be more than one operator in an expression.

To evaluate these types of expressions there is a rule of precedence. It guides the order in which these operations are carried out.

For example, multiplication has higher precedence than subtraction.

But we can change this order using parentheses () as it has higher precedence than multiplication.

Multiplication, division and modulus have the same priority but higher than addition and subtraction.

* / %

+ -

Example Programs

Sum of two Numbers Using hard coded values

```
#include<stdio.h>

int main()
{
    int a=2;

    int b=3;

    int sum=a+b;

    printf("\nSum: %d", sum);

    return 0;
}
```

Output:

Sum: 5

The scanf() function

In C, scanf() is the most commonly used function to take input from user from the standard input device that is keyboard.

Syntax: scanf(format specifier, &variable_name);

Eg: scanf("%d", &age);

Here %d is the format specifier for int data type which means that we are taking an int type variable from user. The & is the reference operator which gives the address of a variable. Here &age means to store the input data on the address of age variable.

Sum of two Numbers by Taking Input from the User

```
#include <stdio.h>

int main() {

    int a, b, result;

    printf("Enter value for a: ");

    scanf("%d", &a);

    printf("Enter value for b: ");

    scanf("%d", &b);

    result=a+b;

    printf("Result is: %d", result);

    return 0;

}
```

Output:

Enter value for a: 2

Enter value for b: 3

Result is: 5

Feet to Inches Converter by Taking Input from the User:

1. `#include <stdio.h>`: This is a preprocessor directive that tells the C compiler to include the standard input-output library (stdio.h). It's necessary for using functions like printf and scanf.
2. `int main()`: This is the main function of the program. Execution of a C program starts from the main function.
Inside the main function:
3. `float feet, inches;`: These lines declare two floating-point variables feet and inches to store the user's input and the converted result.
4. `printf("Enter value for feet: ");`: This line displays a prompt asking the user to enter a value for feet.
5. `scanf("%f", &feet);`: This line uses scanf to read a floating-point number entered by the user and stores it in the feet variable. The & operator is used to get the address of the feet variable for scanf to store the input.
6. `inches = feet * 12;`: Here, it calculates the equivalent number of inches by multiplying the value in the feet variable by 12 (since there are 12 inches in a foot).
7. `printf("%.3f Feet in inches: %.3f", feet, inches);`: This line displays the result using printf. It prints the original value entered for feet and the corresponding value in inches. The format specifier %.3f is used to display floating-point numbers with three decimal places.
8. `return 0;`: This statement indicates the successful termination of the program.

```
#include<stdio.h>

int main()
{
    float feet, inches;
    printf("Enter value for feet: ");
    scanf("%f", &feet);
    inches=feet*12;
    printf("%.3f Feet in inches: %.3f", feet, inches);
    return 0;
}
```

Output:

Enter value for feet: 2.3

2.300 Feet in inches: 27.600