CS-1002 Programming Fundamentals

LAB - 03

Introduction to IDE and Basic Programming Constructs

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES
Fall 2021

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

INTRODUCTION TO THE IDE

- IDE stands for Integrated Development Environment. It is a software application that provides facilities for developing software.
- IDEs increase programmer productivity by combining common activities of writing software into a single application: editing source code, building executables, and debugging.
- For C language programming, we will be using **Dev-C++.**
- Dev-C++ is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language.
- Dev-C++ is free software and is distributed under the GNU General Public License. Thus we can distribute or modify the IDE freely.
- "Bloodshed Software" originally developed it and Orwell has forked it after it was abandoned by Bloodshed in 2006.
- As similar IDEs, it offers to the programmer a simple and unified tool to edit, compile, link, and debug programs.

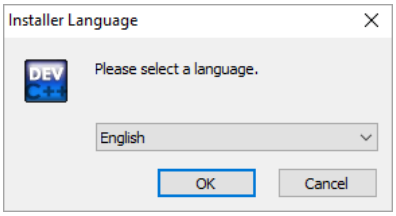
INSTALLATION AND CONFIGURING THE IDE

We can get the appropriate installable for dev-C++ IDE from **https://www.bloodshed.net/**

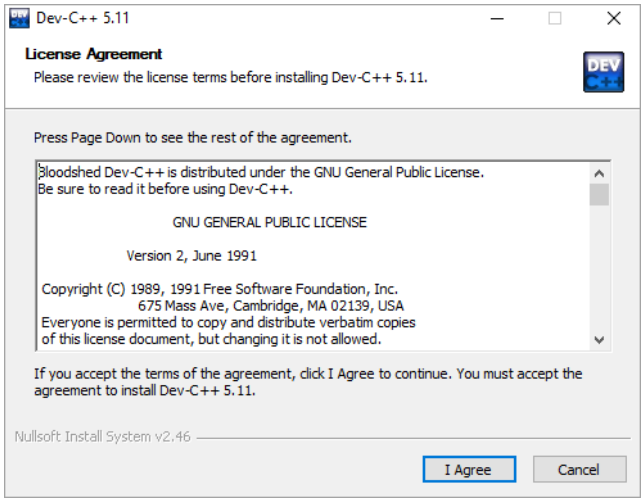
- Let's see the entire installation process now. We have used the installable that comes along with the C++ compiler.

The stepwise installation for Dev-C++ is given below.

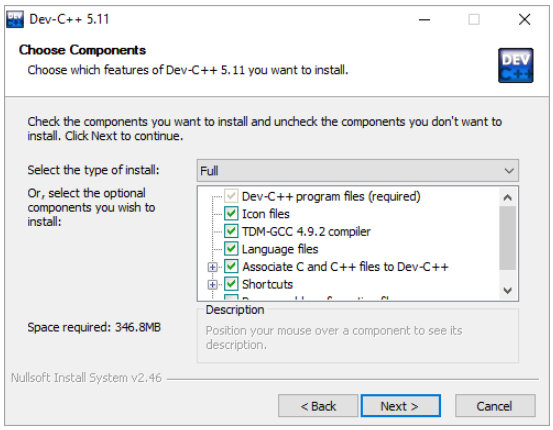
- The first step while we start the installer is to select the language of our choice as shown in the below screenshot.



- Once you select the appropriate language, you have to agree to the license agreement that pop-ups next.

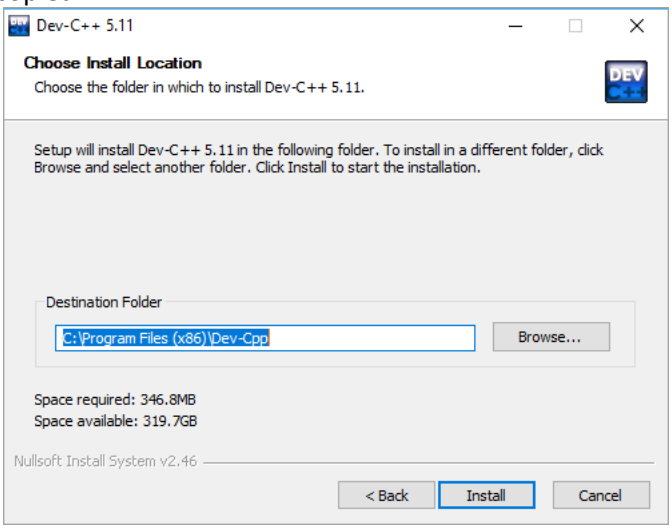


- Next, we are asked to select the components that we need to install as a part of the dev-C++ installation.



As shown in the above screenshot, we are provided with a list of components available for installation and a checkbox against each component. We can check/uncheck each box to indicate which components to install. Click next once the components are selected.

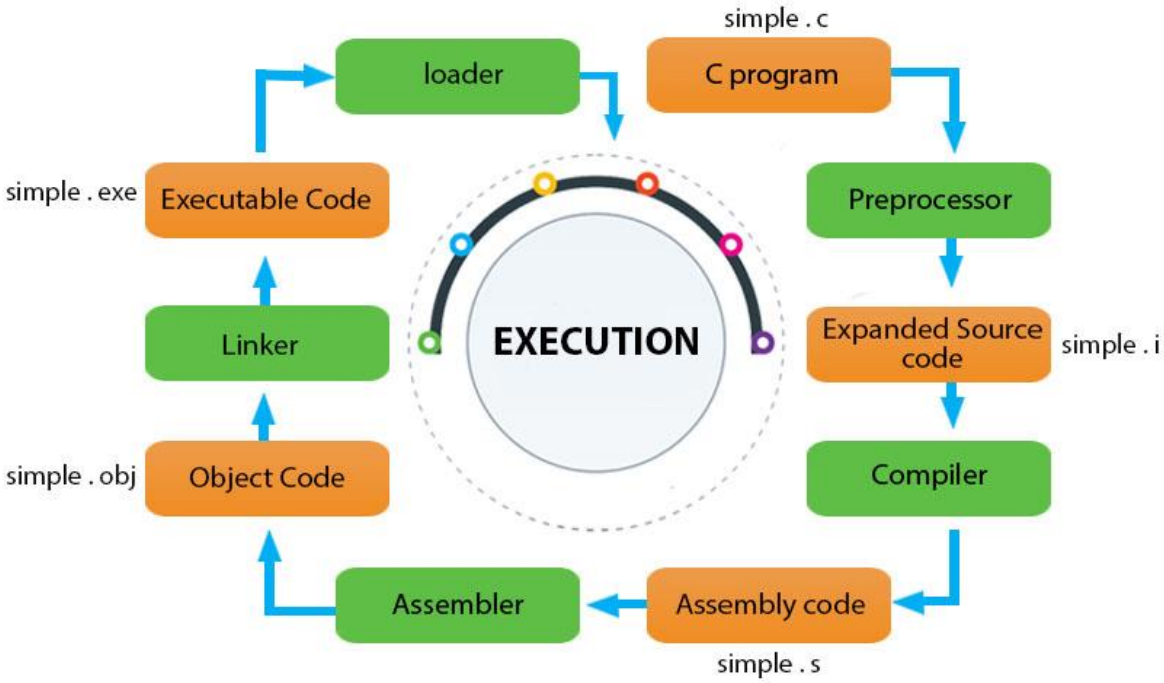
- Now the installer prompts the user for the destination folder where the dev-C++ files/libraries etc. are to be copied.



Once we provide the destination folder path, click on Install.

FLOW OF EXECUTION OF C PROGRAM

- C is a high-level programming language developed in 1972 by Dennis Ritchie at AT&T Bell Laboratories.
- Programs in C typically go through six phases to be executed. These are: edit, preprocess, compile, link, load and execute.



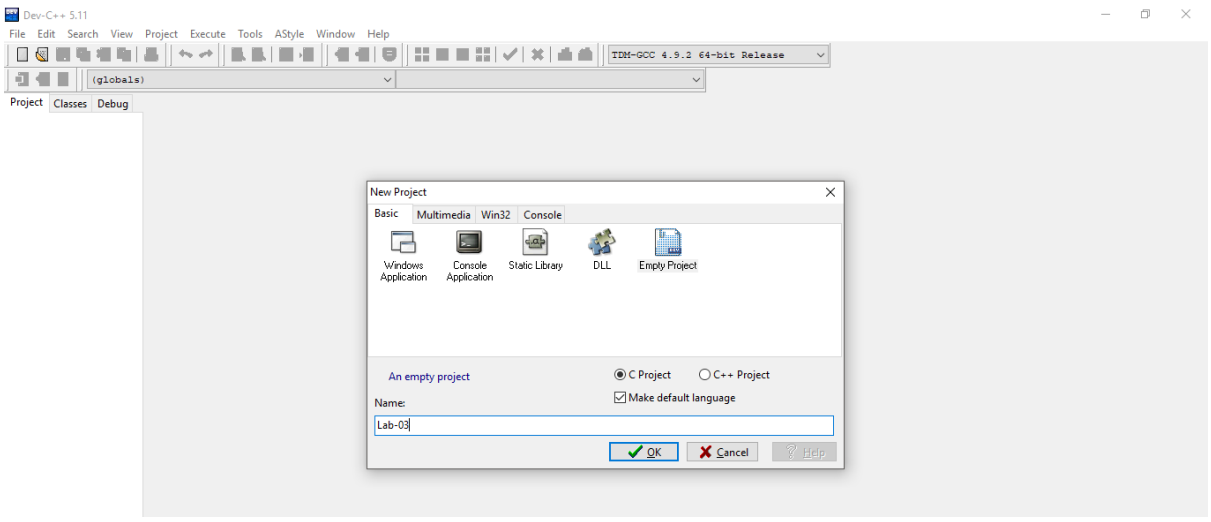
1. The source code written by programmers is stored in the file for example: program.c.
2. C program (source code) is sent to **preprocessor** first. All of these preprocessor directives begin with a '#' (hash) symbol. The '#' symbol indicates that, whatever statement starts with #, is going to the preprocessor program, and preprocessor program will execute this statement. Examples of some preprocessor directives are: #include, #define, #ifndef etc. For example, include will include extra code to your program.
   The preprocessor is responsible to convert preprocessor directives into their respective values. The preprocessor generates an expanded source code.
3. Expanded source code is sent to **compiler** which compiles the code and converts it into assembly code.
   Computer programs are written using high-level programming languages. The source code is converted into machine-understandable machine code. A compiler is used for this conversion. Compiler is a translator that converts the source code from high-level programming language to a lower level machine language in order to create an executable program.

4. The assembly code is sent to **assembler** which assembles the code and converts it into object code. Now a simple.obj file is generated.
5. The object code is sent to **linker** which links it to the library such as header files. Then it is converted into executable code.
   Library functions are not a part of any C program but of the C software. Thus, the compiler doesn't know the operation of any function, whether it is printf or scanf. The definitions of these functions are stored in their respective library which the compiler should be able to link. This is what the Linker does. So, when we write #include, it includes stdio.h library which gives access to Standard Input and Output. The linker links the object files to the library functions and the program becomes a .exe file. A simple.exe file is generated which is in an executable format.
6. The executable code is sent to **loader** which loads it into memory and then it is executed. After execution, output is sent to console.
   Whenever we give the command to execute a particular program, the loader comes into work. The loader will load the .exe file in RAM and inform CPU to execute a particular program.

DEVELOPMENT USING DEV-C++ IDE

CREATE A NEW PROJECT

- To create a new project or source file in dev-C++ we need to follow the below steps:
- Click File -> New->Project
- Here, we can specify the project name. Make sure to select the "Empty Project" and also to check the "C Project" button.



- Once the entire information is provided, we can click ok and the IDE will ask for the path where the project is to be saved. When this is done, a workspace will open with the project explorer on the left-hand side that shows the project we just created.

ADD A SOURCE FILE TO A PROJECT

- Add a new file by clicking **Project ->New File** or Right-click on **Project Name** in the project explorer and click **New File**
- Save the file using CTRL+S or save option in file menu give it a name and choose extension c

HELLO WORLD PROGRAM

- This is the traditional "Hello World" program in C language

```c
#include <stdio.h>

main(){

    printf("Hello World!");

}
```

Understanding HELLO WORLD Program

- This program uses (that is, it 'includes') code from the C-language 'standard input/output library, stdio, using this statement:

#include <stdio.h>

- The code that starts with the name main is the 'main function' – in other words, it is the first bit of code that runs when the program runs. The function name is followed by a pair of parentheses. The code to be run is enclosed between a pair of curly brackets:

main() {

}

- In this case, the code calls the C printf function to print the string (the piece of text) between double-quotes.

printf("hello world");

COMPILE/BUILD

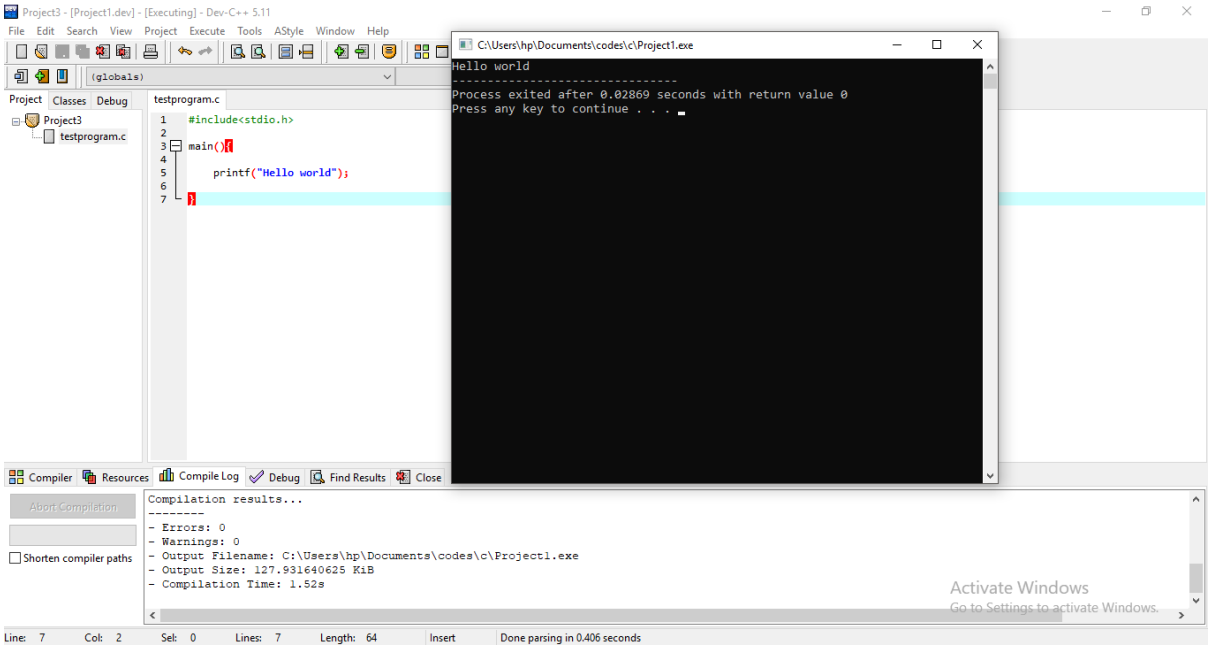
When we have all the code ready for the project, we will now compile and build the project.

Follow the below steps to build and execute the dev C++ project:

- To compile the project, click **Execute -> Compile** (or click F9).
- We can see the compilation status in the "**Compile Log**" tab in the workspace.
- If there are any errors whether syntax or linker errors, then they will appear in the compiler tab.
- Once the project is compiled successfully, we need to run it.

EXECUTE PROJECT

- Click on **Execute ->Run**.(or click F10)
- The console window that gives us the output will be shown in the below screenshot.



COMPILING A C PROGRAM WITHOUT IDE

We usually use a compiler with a graphical user interface, to compile our C program. This can also be done by using cmd.

STEP 1: Run the command 'gcc -v' to check if you have a compiler installed. If not you need to download a gcc compiler and install it. You can search for cmd in your windows system to open the command prompt.

Step 2: Create a c program and store it in your system.

STEP 3: Change the working directory to where you have your C program. You can do that by using the command '**cd**', which changes the directory. We need to pass the name of the directory in which the program is stored.

Example: cd Desktop.

Our program is already in the user directory so we don't have to change it.

STEP 4:

The next step is to compile the program. To do this we need to use the command gcc followed by the name of the program we are going to execute. In our case, we will use p2.c. Note that p2.c is the name of the file.

```
C:\Users\hp\Documents\codes\c>gcc p2.c -o p2
```

With –o we can set exe file name. Here the exe file will have name p2. If you don't set this name then by default compiler will generate exe file with name a.exe

After this, an executable file will be created in the directory that your c file exists in. Eg: p2.exe

STEP 5:

In the next step, we can run the program. This is done by simply giving the name of the executable file without any extension. On giving this we will get an output. Here, our p2 code is executed and we got output for this code.

```
C:\Users\hp\Documents\codes\c>p2
Hello world
```

DATATYPES IN C

- In C programming, data types are declarations for variables. This determines the type and size of data associated with variables.

Here's a table containing commonly used types in C programming for quick access.

Data Type	Description	Size(bytes)
int	Integers are whole numbers that can have both zero, positive and negative values but no decimal values. It can take 232 distinct states from -2147483648 to 2147483647.	at least 2, usually 4
float	Floating type variables can hold real numbers precision of 6 digits	4
double	Floating type variables can hold real numbers with precision of 14 digits	8
char	Character data type allows a variable to store only one character.	1

VARIABLES IN C

- In programming, a variable is a container (storage area) to hold data.
- To indicate the storage area, each variable should be given a unique name (identifier).
- Variable names are just the symbolic representation of a memory location.

Syntax:
A variable can be declared and initialized using following syntax:

```
datatype variable_name = value ;
```

Example:

```
int playerScore = 95;
```

Here, playerScore is a variable of int type. Here, the variable is assigned an integer value 95

Rules for naming a variable

- A variable name can only have letters (both uppercase and lowercase letters), digits and underscore.
- The first letter of a variable should be either a letter or an underscore.
- There is no rule on how long a variable name (identifier) can be. However, you may run into problems in some compilers if the variable name is longer than 31 characters.

Note: You should always try to give meaningful names to variables. For example: firstName is a better variable name than fn.

- C is a strongly typed language. This means that the variable type cannot be changed once it is declared.

FORMAT SPECIFIERS

- Format Specifiers are strings used in the formatted input and output.
- The format specifiers are used in C to determine the format of input and output.
- Using this concept the compiler can understand that what type of data is in a variable while taking input using the scanf() function and printing using printf() function.

Some of the commonly used Format Specifiers are given below:

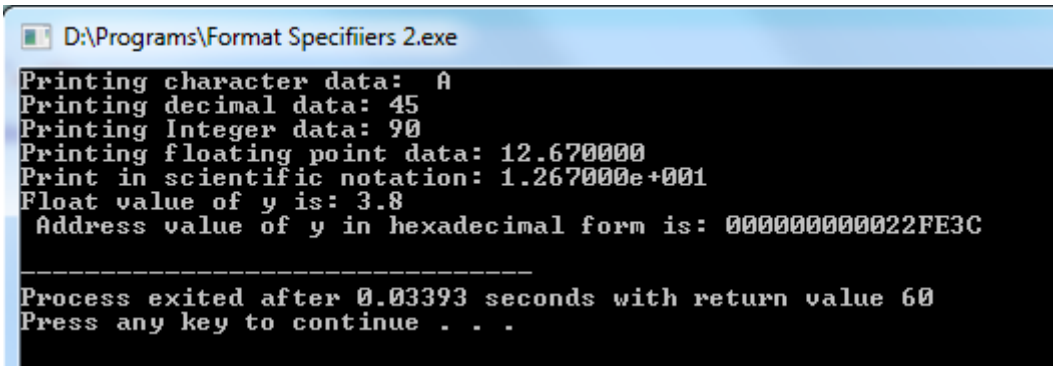
Format specifier	Description
%d or %i	It is used to print the **signed integer value** where signed integer means that the variable can hold both positive and negative values.
%u	It is used to print the **unsigned integer value** where the unsigned integer means that the variable can hold only positive value.
%f	It is used for printing the **decimal floating-point values**. By default, it prints the 6 values after '.'
%e / %E	It is used for **scientific notation**. It is also known as Mantissa or Exponent.
%g	It is used to print the **decimal floating-point values**, and it uses the fixed precision, i.e., the value after the decimal in input would be exactly the same as the value in the output.
%c	It is used to print the **unsigned character**.

%s	It is used to print the **strings**.
%ld	It is used to print the **long-signed integer value**.

Example:

```c
#include <stdio.h>

main() {
//printing character data
   char ch = 'A';
   printf("Printing character data:  %c\n", ch);

   //print decimal or integer data with d and i
   int x = 45, y = 90;
   printf("Printing decimal data: %d\n", x);
   printf("Printing Integer data: %i\n", y);

   //print float value
   float f = 12.67;
   printf("Printing floating point data: %f\n", f);

   //print in scientific notation
   printf("Print in scientific notation: %e\n", f);

  float z=3.8;
  printf("Float value of y is: %g\n ", z);

  printf("Address value of y in hexadecimal form is: %p\n", &y);


}
```

Output



OUTPUT IN C

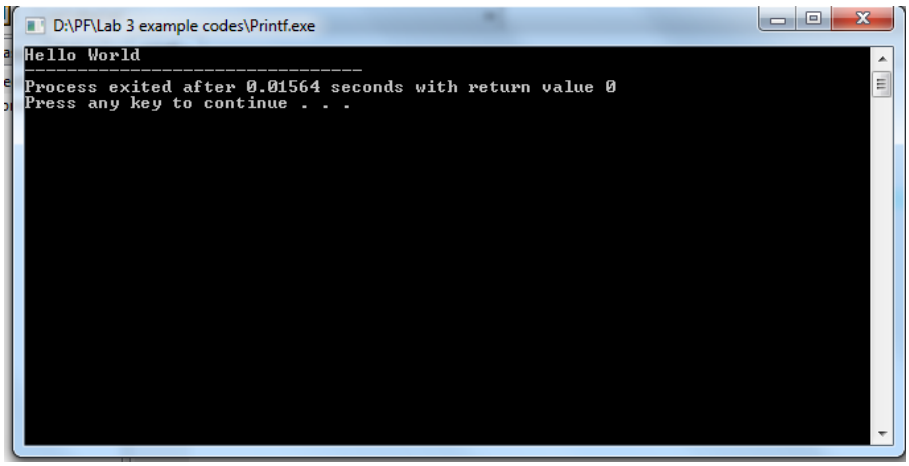
- In C programming, printf() is one of the main output function. The function sends formatted output to the screen.
- The syntax of printf() function is given below:

printf("format specifier",variable_name);

Example

```c
#include <stdio.h>

int main(){

    printf("Hello World");

}
```

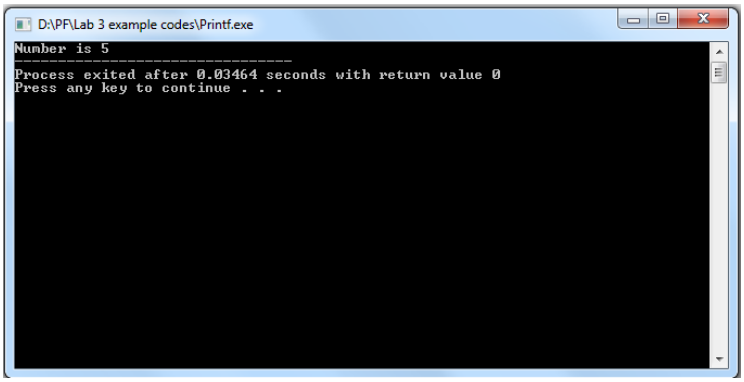
Output



Example # 2:

```c
#include <stdio.h>

int main(){

    int testInteger=5;
    printf("Number is %d",testInteger);

}
```

Output



We use %d format specifier to print int types. Here, the %d inside the quotations will be replaced by the value of testInteger

INPUT IN C

- In C programming, scanf() is one of the commonly used function to take input from the user.
- The scanf() function reads formatted input from the standard input such as keyboards.
- The syntax of printf() function is given below:

scanf("format specifier",variable_name);

Example:

testprogram.c

```c
#include <stdio.h>
int main()
{
    int testInteger;
    printf("Enter an integer: ");
    scanf("%d", &testInteger);
    printf("Number = %d",testInteger);
    return 0;
}
```

Output

```
Enter an integer: 2
Number = 2
--------------------------------
Process exited after 9.611 seconds with return value 0
Press any key to continue . . .
```

Example # 2: getting multiple inputs

```c
#include <stdio.h>
int main()
{
    int a;
    float b;

    printf("Enter integer and then a float: ");

    // Taking multiple inputs
    scanf("%d%f", &a, &b);

    printf("You entered %d and %f", a, b);
    return 0;
}
```

Output

```
Enter integer and then a float: 4
5.5
You entered 4 and 5.500000
--------------------------------
Process exited after 8.889 seconds with return value 0
Press any key to continue . . .
```

ESCAPE SEQUENCE

- Escape Sequences are non-printing characters.
- When a character is preceded by a backslash (\), it is called an escape sequence and it has a special meaning to the compiler.

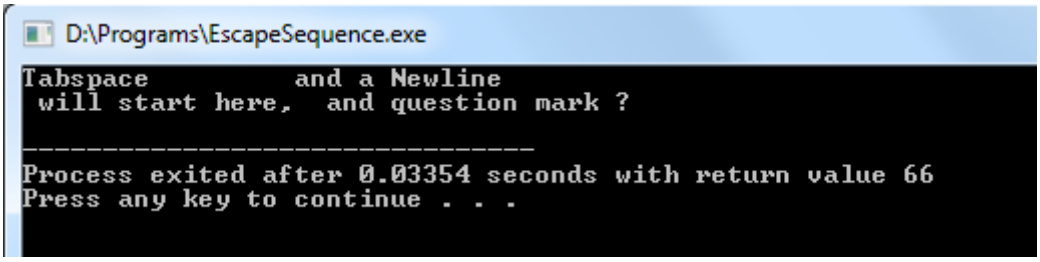
Escape Sequence & Description

	\t **Inserts a tab in the text.**
	\b **Inserts a backspace in the text.**
	\n **Inserts a newline in the text.**
	\r **Inserts a carriage return in the text.**
	\' **Inserts a single quote character in the text.**
	\" **Inserts a double quote character in the text.**
	**** **Inserts a backslash character in the text.**
	\? **Inserts a question mark in the text.**
	\a **Play beep or Alarm**

Example

```
#include <stdio.h>
int main() {
   char ch1='\t';
   char ch2 = '\n';
printf( "Tabspace %c and a Newline %c will start here,  and question mark \? \n", ch1, ch2);

}
```

Output:



PRECISION SETTING IN C

Precision is specified by the number of digits after the decimal point for the outputs for float as well as double numbers.

If precision is not specified, it would be according to the default setting in the computer which is generally 6 digits.

The precision may be specified in the format specifiers place by a period(.) followed by a positive number equal to the number of digits desired.

Example:

```c
#include<stdio.h>

int main(){
    double a=2.55555588889999;

    printf("before setting the precision\nnumber is: %lf",a);

    printf("after setting the precision\nnumber is: %.14lf",a);
}
```

Output:

```
before setting the precision
number is: 2.555556after setting the precision
number is: 2.55555588889999
--------------------------------
Process exited after 0.04312 seconds with return value 55
Press any key to continue . . .
```

The above will display 14 digits after the decimal point after setting the precision otherwise it will display only 6 characters even though double datatype precision is 14 digits as seen in datatypes section.

LAB#01 EXERCISES

QUESTION#1

Write the datatypes of the following values:

- “Introduction to IDE and basic programing constructs”
- 3.14
- 300000000
- 7.0000000000005

QUESTION#2

Explain the output of this C program. Why the wrong value is being displayed in the output?

```c
#include <stdio.h>

int main(){

    int testInteger=3000000000;
    printf("Number is %d",testInteger);

}
```

```
D:\PF\Lab 3 example codes\Printf.exe
Number is -1294967296
--------------------------------
Process exited after 0.03059 seconds with return value 0
Press any key to continue . . .
```

QUESTION#3

Write a C program that takes two integer values as input from the user. Then swap the values taken from the user and display the output of the variables.

QUESTION#4

A customer asks the IT firm to develop a program in C language, which can take tax rate and salary from the user on runtime and then calculate the tax, the user has to pay and the salary he/she will have after paying the tax. Tax rate must be between 0-100.

QUESTION#5

A car traveled for some hours. The time car traveled is taken at run time of the program, and it must not be negative and must be between **one to five hours**. The car had not traveled same distance in each hour. The distance that the car covered must not be negative. Write a C Program that computes the Average Speed of the Car in miles per hour. Hint: the restrictions can be displayed in the form of message on the window.

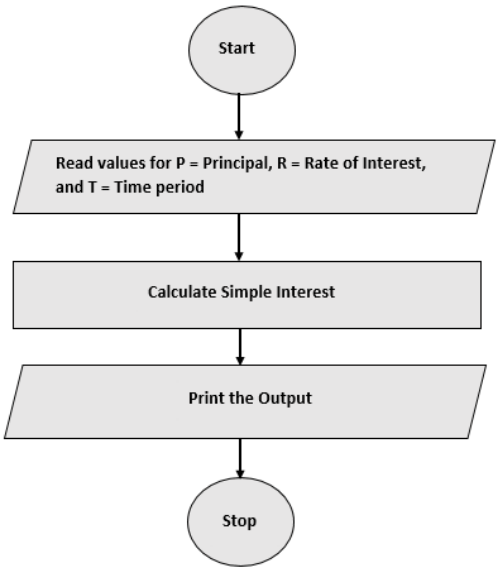
QUESTION#6

A person travels 10 kilometers in south from his home to go to the university in car then he travels 10 kilometers in west then he travels another 5 kilometers south and finally he travels 15 kilometers in north with the fuel consumption of 3 liters/km. Calculate how far away is his university from home and how much fuel he consumes daily in this two way trip?

QUESTION#7

Construct a C program with the flowchart below. The input value of the Principle must be between 100 Rs. To 1,000,000 Rs. The Rate of interest must be between 5% to 10% and Time Period must be between 1 to 10 years. Hint: these restrictions can be displayed in the form of message on the window. Simple interest can be computed by using the formula mentioned below.

Simple Interest=(P*R*T)/100



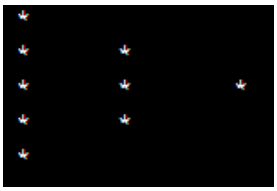
QUESTION#8

Calculate the area for a rectangle in which user inputs width and length of a rectangle. **Formula:** area of a rectangle=width * height.

QUESTION#9

Write a C program to print the following shapes using escape sequences. Moreover, you are required to compile and execute these program using CMD

a) 

b) 

QUESTION#10

For 4 months, Ali starts a diet plan to lose his weight. In the first month, he loses 5kg. In the second month he loses another 10kg, In the third moth he loses another 12kg and finally in the last month he loses 7kg. calculate how much weight he has lost per month, on average? Also calculate his weight right now if his initial weight was 110kg?