Aror University of Art, Architecture, Design & Heritage Sukkur.

Department of Artificial Intelligence and Multimedia Gaming

Fundamentals of Programming (Fall-2025)

LAB No. 1

Prepared by: Abdul Haseeb Shaikh

Objective of Lab No. 1:

After performing lab1, students will be able to:

- Install DEV C++
- Understand the basic terminology of Programming
- Configure DEV C++ Environment
- Create new Program or Project
- Write Hello World program using C++ syntax
- Understand the Hello World program
- Use the comments in their programs

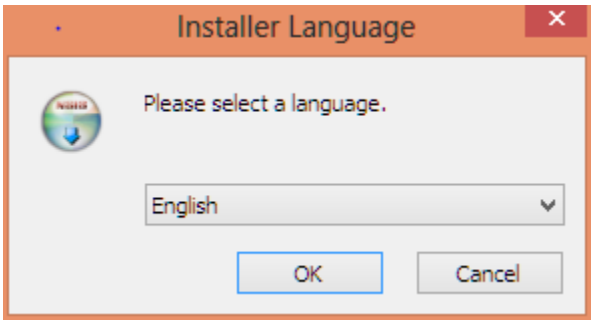
Installing DevC++:

DevC++ is a free integrated development environment for Windows operating system which is distributed under GNU general public license for programming in C and C++. An integrated development environment provides comprehensive facilities to programmers. It normally consists of source code editor, build automation tools and debugger. Most modern IDEs offer intelligent code completion features. Follow the below given steps to install the DevC++.

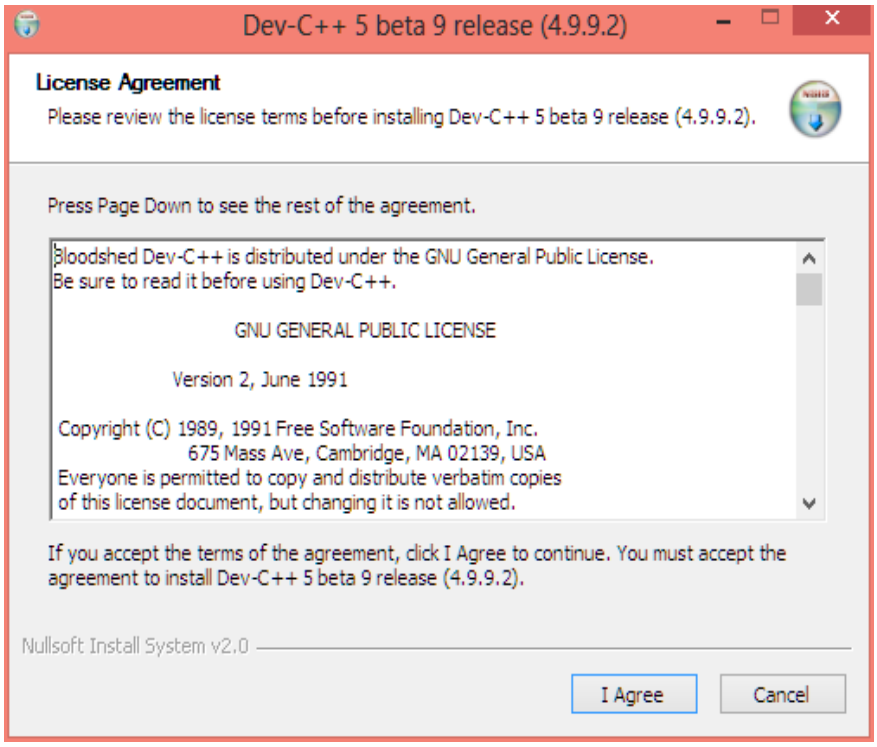
1. To install DevC++, you first need DevC++ setup. You can download the DevC++ setup from the course yahoo group. If you are not still the member of our yahoo group then use (http://sourceforge.net/projects/dev-cpp/files/Binaries/Dev-C%2B%2B%204.9.9.2/devcpp-4.9.

9.2_setup.exe/download?use_mirror=kaz) link to download the DevC++ setup file.

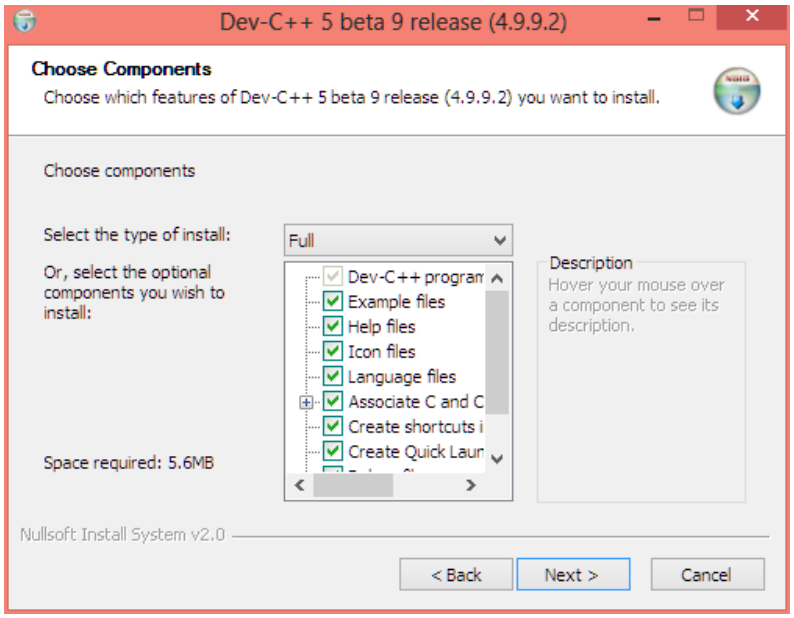
2. Open the setup file by double clicking on it. The following dialog box will appear, select the language from the list and click OK.



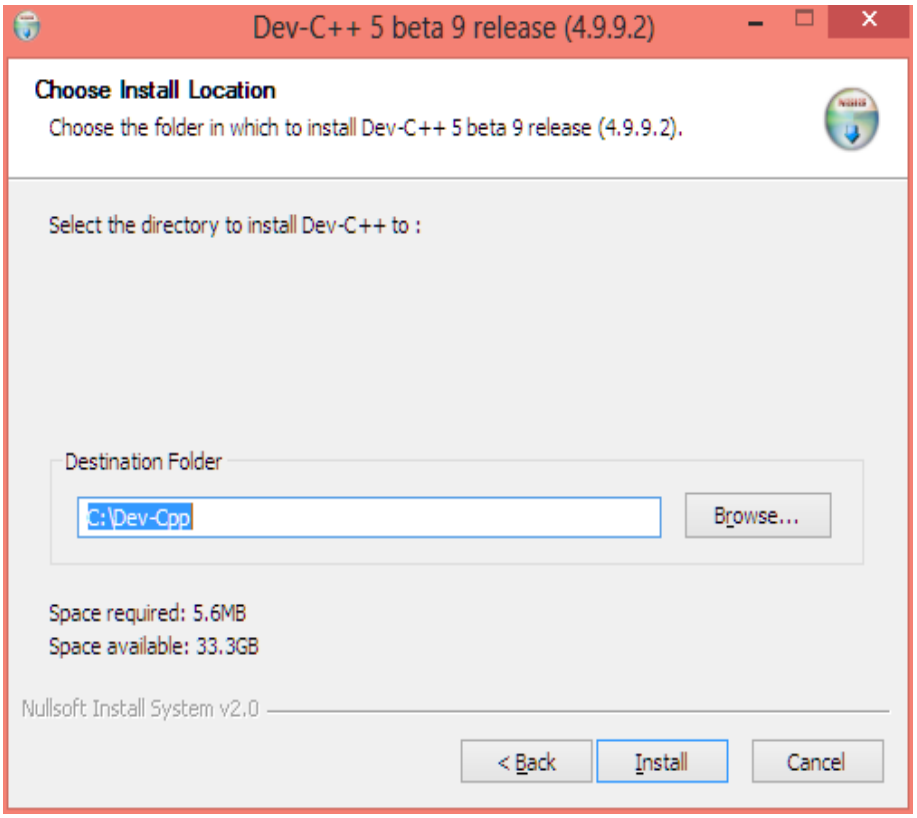
3. In the next step, the setup will ask you to review the license terms before installing Dev-C++. Read the license terms, and click "I Agree" button if you agree to the terms and conditions.



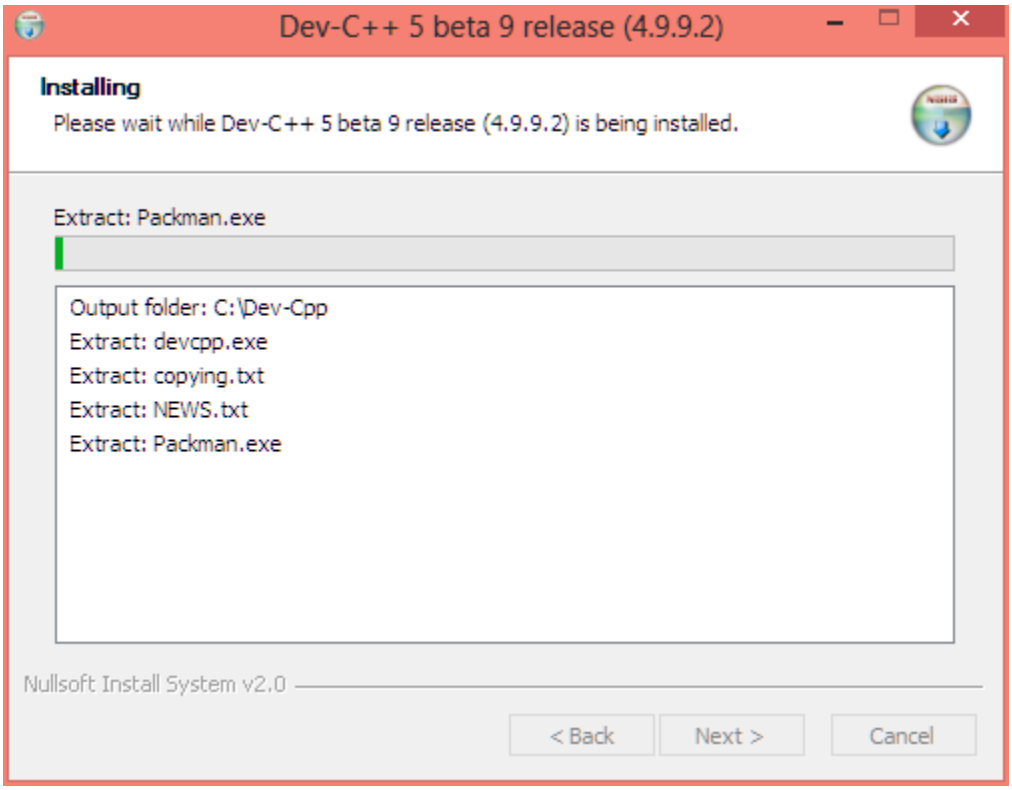
4. After step 3, the setup will ask you to choose the type of installation or the features of Dev-C++ you want to install, by default the installation type is set to "Full" and all features are selected. Click on "Next" button.



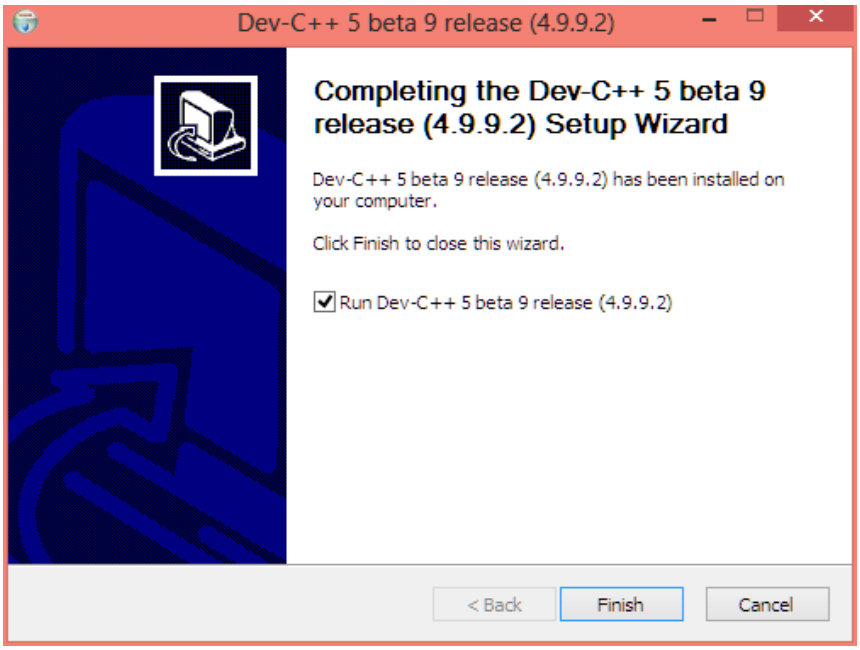
5. After selecting features, select the location where the Dev-C++ will be installed. By default, the location is set to "C:\Dev-C++", click on "Install".



6. setup will install the Dev-C++ as shown in the following screen.



7. When installation is complete, the following screen appears, click on "Finish" to exit the setup and run Dev-C++.



Configuring DevC++ Environment:

Open DevC++ from start menu and go to

1. Tools□Compiler Options□Settings□Linker□Set Generate Debugging information bit to ***Yes*** from ***No***.
2. Tools□Environment Options□Files and Directories tab□set user default directories settings (Here your project file will be saved by default).
3. Tools□Editor Options□Snippets□Default Source □Add following default code □Check on insert default code check□Ok (Whenever you create a new file you will see following default code in that file. We will discuss this code in subsequent sections in this lab)

```
#include <iostream>
using namespace std ;
int main ()
```

```
{

system ("PAUSE") ;
return 0 ;
}
```

Creating a new program or project:

For creating a new program you have to either open a new source file or new project. If you want to create a single file the best choice is to open a new source file. If you want to create a project which may have multiple files then opening a new project is good choice.

Opening a new source file:

Go to file□New□Source File□Save the source file with the meaningful name□Start writing your code

Opening a new project:

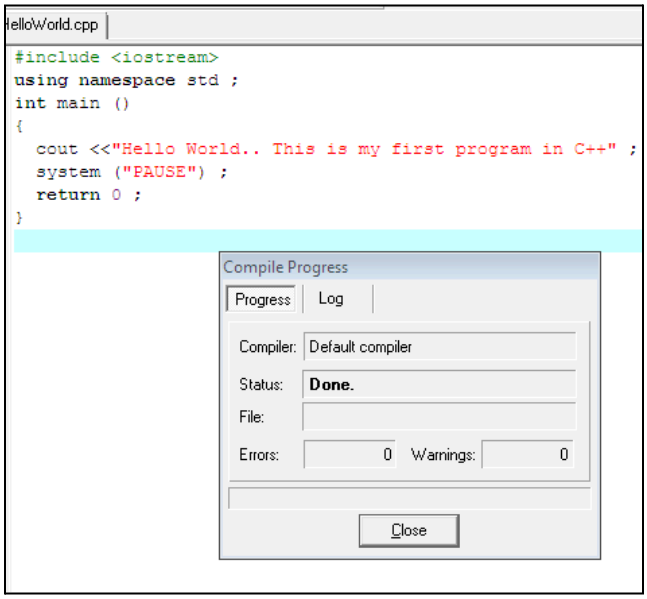
Go to file□New□Project□Create the folder□save project inside the folder□Also save main file inside the folder□Ok

You can add more files in the project.

For now open a new source file, save it with the name of HELLOWORLD and type following program.

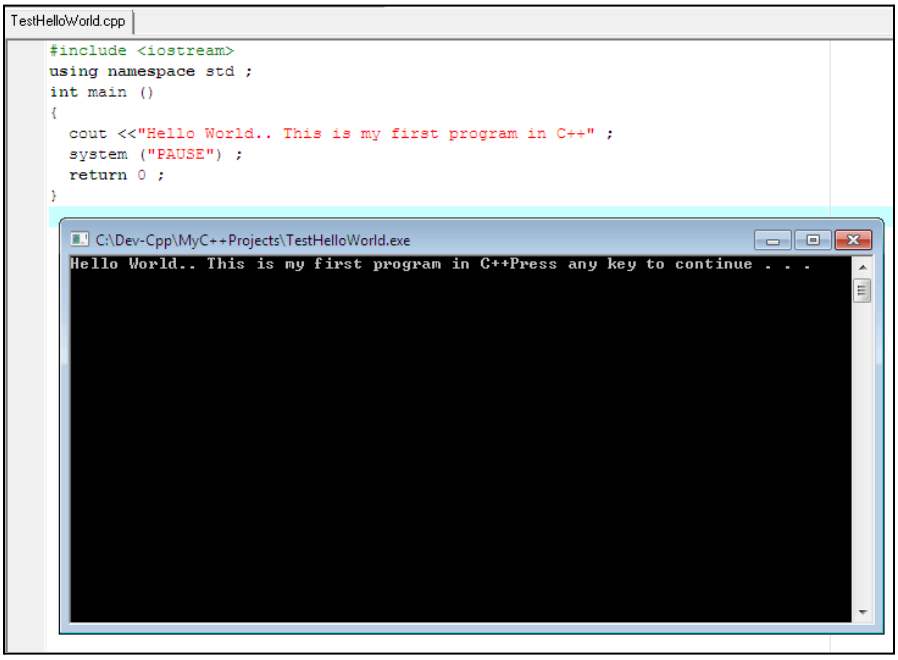
```
#include <iostream>
using namespace std ;
int main ()
{
 cout <<"Hello World.. This is my first program in C++" ;
 system ("PAUSE") ;
 return 0 ;
}
```

Compile the program by pressing CNTRL+F9 or click on Execute menu and press compile. Successful compilation will show following screenshot.



Now to see the result you need to execute or run this program by pressing CNTRL+F10 or click on Execute menu and press Run. Successful execution will show following screenshot.

Note: If you want to compile and run program simultaneously then the good option is to press F9 or you can also click Execute menu and press compile and run option.



Hurrah! So you have successfully written the first program in C++ using DevC++ IDE.

Understanding the program:

#include <iostream>:

Lines beginning with a hash sign (#) are directives read and interpreted by what is known as the *preprocessor*. They are special lines interpreted before the compilation of the program itself begins. In this case, the directive #include <iostream>, instructs the preprocessor to include a section of standard C++ code, known as *header iostream*, that allows to perform standard input and output operations, such as writing the output of this program (Hello World) to the screen.

using namespace std ;

int main ():
This line initiates the declaration of a function. Essentially, a function is a group of code statements which are given a name: in this case, this gives the name "main" to the group of code statements that follow. Functions will be discussed in detail in a later chapter, but essentially, their definition is introduced with a succession of a type (`int`), a name (`main`) and a pair of parentheses (`()`), optionally including parameters. The function named `main` is a special function in all C++ programs; it is the function called when the program is run. The execution of all C++ programs begins with the `main` function, regardless of where the function is actually located within the code.

{:
Opening curly bracket of main function

cout:
It is the output stream. Whenever we want to display the message on screen we use cout stream

<<:
It is the insertion operator. It is always used with cout.

"Hello World.. This is my first program in C++":
It is the message which we want to print on screen.

;:
It is the termination point of instruction. And any line which ends at ; is called the instruction or statement.

system ("PAUSE") ;:
It is function. We use this to hold output screen
return 0 ;:
Return is a keyword. Because the return type of main is int so we need to return any value hence we are returning an integer value
}:

Closing curly bracket of main function

Task 01: Modify your Program, use: cout>>"Hello World"; this time, and check the Output.

Task 02: Modify your Program, use: cout "Hello World"; this time, and check the Output.

Task 03: Modify the Program, ask it to print your name

Task 04: Print following text on screen:

```
Welcome to Lab 1
Bye
--------------------------------
Process exited after 0.05294 seconds with return value 0
Press any key to continue . . .
```

Task 05q: Use single Line and Multiline comments to add clarity to your code

A. **Add a Multiline comment at the top of your file, indicating your Roll No, Your Name, Lab_Number etc in following manner:**

   Roll No: 01
   Name: Ahmed
   Lab_Number: 01

B. **Add a Single Line Comment wherever necessary.**