



**Institute of Computer Sciences & Information Technology (ICS&IT)
Faculty of Management and Computer Science (FMCS)
The University of Agriculture, Peshawar**

Lab No. 2: Introduction to Computer Programming

Objective:

At the end of this lab students will know about

- Introduction to Dev-C++ IDE (Integrated Development Environment)
- Simple C++ program in DEV
- Understand Program
- Input and Output Statements
- Types and Uses of escape sequences

What is C++?

C++ is a cross-platform language that can be used to create high-performance applications.

C++ was developed by Bjarne Stroustrup, as an extension to the C language.

C++ gives programmers a high level of control over system resources and memory.

The language was updated 3 major times in 2011, 2014, and 2017 to C++11, C++14, and C++17.

Why Use C++

C++ is one of the world's most popular programming languages.

C++ can be found in today's operating systems, Graphical User Interfaces, and embedded systems.

C++ is an object-oriented programming language which gives a clear structure to programs and allows code to be reused, lowering development costs.

C++ is portable and can be used to develop applications that can be adapted to multiple platforms.

C++ is fun and easy to learn!

As C++ is close to C# and Java, it makes it easy for programmers to switch to C++ or vice versa

Introduction to Dev-C++ IDE

An IDE (Integrated Development Environment) is used to edit AND compile the code.

Popular IDE's include Code::Blocks, Eclipse, and Visual Studio. These are all free, and they can be used to both edit and debug C++ code.

Dev-C++ is a full-featured Integrated Development Environment (IDE) for the C/C++ programming language. As similar IDEs, it offers to the programmer a simple and unified tool to edit, compile, link, and debug programs.

Dev-C++ is a Free Software distributed. The IDE can be downloaded here:

<https://sourceforge.net/projects/orwelldevcpp/>

First steps

The application development process encompasses the following steps:

1. Create a project

The type of application and the programming language to be used are specified.

2. Write source code

Write the program in C and save the source code file.

3. Compile and link the code

The source code is compiled and linked to generate a running program. Other files of the project may be created.

4. Fix compilation errors, if any

If the syntax of the program is not correct, the compilation fails and the compiler generates a message related to the error/s. The programmer must correct the errors.

5. Run the program

Run the program to validate the functioning.

6. Fix execution errors, if any

If the actions performed by the program are not as expected, it is necessary to correct the source code. It may be also convenient to use the debugger to find complex errors.

Steps

a) Start Dev-C++

Start the IDE from the Program folder Dev-C++ or from the shortcut on desktop.

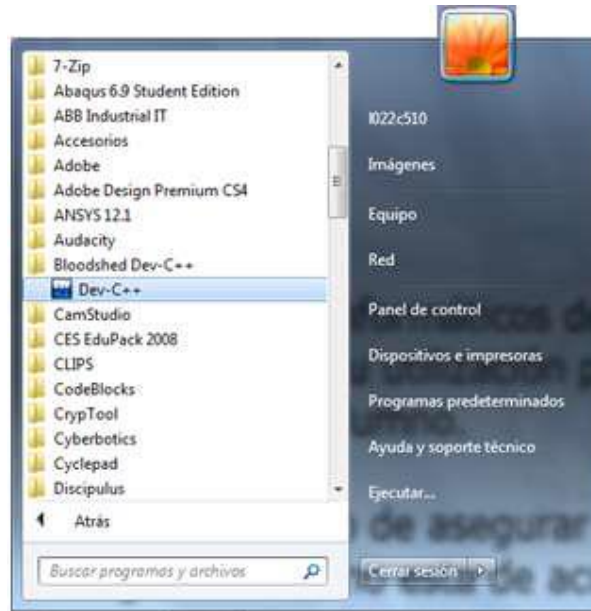


Figure 1. Running Dev-C++ in the computer lab

b) Create a new source file

File ->New->Source File

Or

Ctrl+N

c) IDE

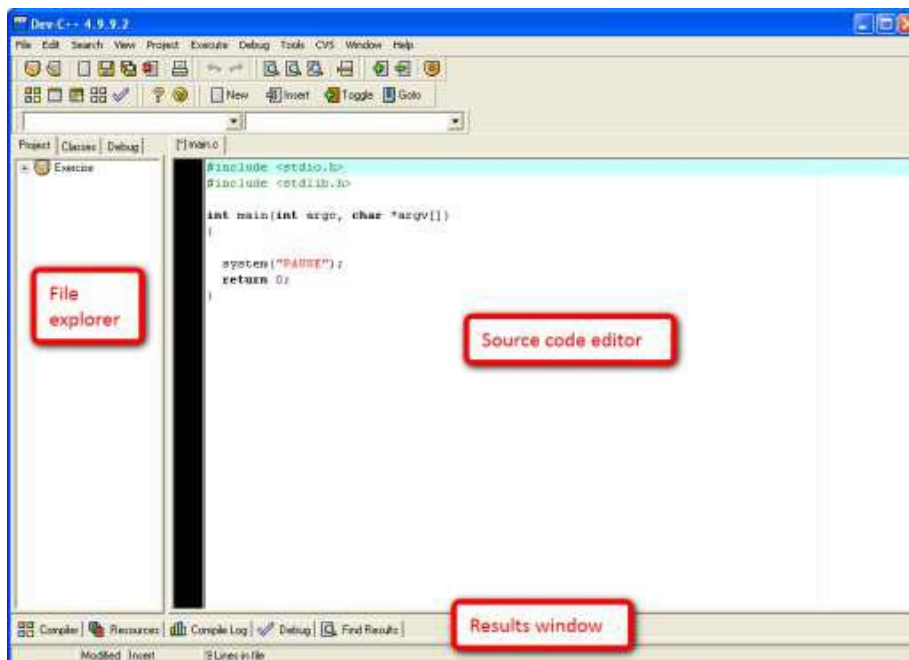
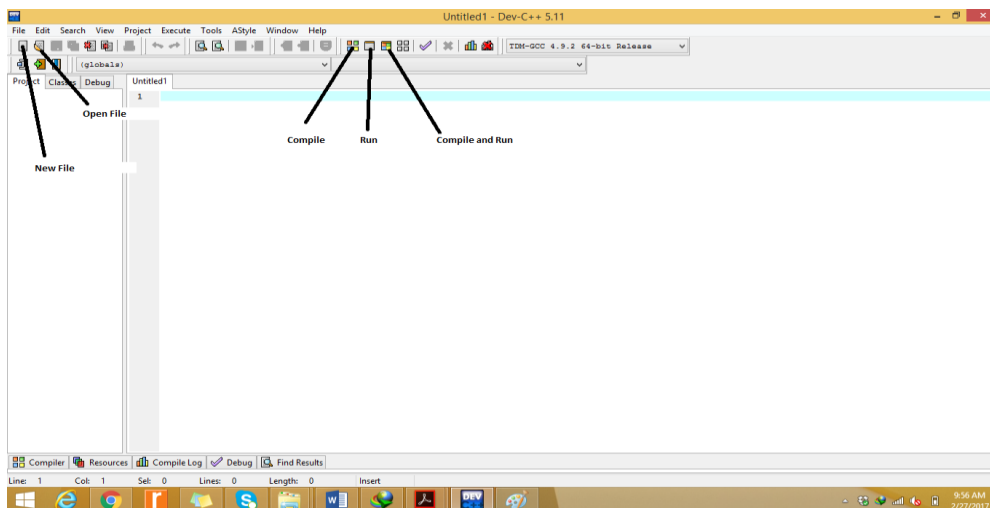


Figure 02. Dev-C++ launches after project creation

- d) Write code and Save file as Program_name.cpp
- e) For Compiling and Running program



Computer is organized in different units in which the basic units are input, output, memory and CPU. Input unit provide data and instructions to the CPU. Memory stores the data and instructions; CPU executes the instructions and pass the results of the execution to the output. Algorithm is a well-define and ordered set of instructions which lead to a solution within finite number of steps.

Compiler is a program that translates high-level language to machine language and creates an executable file. C++ program has one function main () by default without which the program will not execute as the execution starts from the first instruction of the main function. In C++ taking input from user and printing output on display of computer is done by including a header file in the C++ code called iostream which notifies the preprocessor to include the input/output stream contents.

Simple C++ program in DEV and understanding.

```
//Hello World
#include <iostream>    // Pre-processor Directives
using namespace std;
int main()    // Body of Program
{
    cout << "Hello World";
    retrun 0;
}
```

Example Explained

Line 1: `#include <iostream>` is a **header file library** that lets us work with input and output objects, such as `cout` (used in line 5). Header files add functionality to C++ programs.

Line 2: `using namespace std` means that we can use names for objects and variables from the standard library.

Don't worry if you don't understand how `#include <iostream>` and `using namespace std` works. Just think of it as something that (almost) always appears in your program.

Line 3: A blank line. C++ ignores white space.

Line 4: Another thing that always appear in a C++ program, is `int main()`. This is called a **function**. Any code inside its curly brackets `{ }` will be executed.

Line 5: `cout` (pronounced "see-out") is an **object** used together with the INSERTION OPERATOR (`<<`) to output/print text. In our example it will output "Hello World".

Note: Every C++ statement ends with a semicolon `;`.

Note: The body of `int main()` could also been written as:

```
int main () { cout << "Hello World! "; return 0; }
```

Remember: Compiler ignores white spaces. However, multiple lines makes code more readable.

Line 6: `return 0` ends the main function.

Line 7: Do not forget to add the closing curly bracket `}` to actually end the main function.

C++ Comments

Comments can be used to explain C++ code, and to make it more readable. It can also be used to prevent execution when testing alternative code. Comments can be singled-lined or multi-lined.

Single-line comments start with two forward slashes (`//`). Any text between `//` and the end of the line is ignored by the compiler (will not be executed). This example uses a single-line comment before a line of code:

Example

```
// This is a comment  
cout << "Hello World!";
```

[Run example »](#)

This example uses a single-line comment at the end of a line of code:

Example

```
cout << "Hello World!"; // This is a comment
```

C++ Multi-line Comments

Multi-line comments start with `/*` and ends with `*/`.

Any text between `/*` and `*/` will be ignored by the compiler:

Example

```
/* The code below will print the words Hello World!  
to the screen, and it is amazing */  
cout << "Hello World!";
```

C++ Input and Output

cout is used to output (print) values and **cin** is used to get user input. **cin** is a predefined variable that reads data from the keyboard with the extraction operator (`>>`) and **cout** is predefined to denote an output stream that goes to the standard output device (display screen) with the insertion operator (`<<`)

In the following example, the user can input a number, which is stored in the variable x. Then we print the value of x:

Example

```
int x;  
cout << "Type a number: "; // Type a number and press enter  
cin >> x; // Get user input from the keyboard  
cout << "Your number is: "<< x; // Display the input value
```

Escape Sequences

A symbol used to represent a special character in a text literal. The `\n` indicates a newline character. It is an example of an escape sequence. Here are all the C++ escape sequences which you can include in strings:

Escape Sequence	Represented Character
<code>\a</code>	System bell (beep sound)
<code>\b</code>	Backspace
<code>\f</code>	Form feed (page break)
<code>\n</code>	Newline (line break)
<code>\r</code>	"Carriage return" (returns cursor to start of line)
<code>\t</code>	Tab
<code>\\</code>	Backslash
<code>\'</code>	Single quote character
<code>\"</code>	Double quote character

TASKS:

1. Write a C++ code that displays your name, department, and section on the console. Make sure the three things are in three different lines (Use escape sequence \n)
2. Write a C++ code that displays the name and quote of your favorite personality. Make sure to put the quote in inverted commas. (Use escape sequence \" double quotation mark)

Example of output:

Steve Jobs once said, "The doers are the major thinkers. The people that really create the things that change this industry are both the thinker and doer in one person."

3. Print the below patterns? (Use escape sequence)

```
*****  
*****  
*****  
**
```

4. Make your CV that includes your Name, Father's Name, CNIC, Qualification, Semester, CGPA etc.? Print each line separately using "\n" and "endl"? Use comments also.
5. Describe different types of computer languages (at least 8).
6. Print Diamond using escape sequence "\n" and "\t"? In single cout statement?

Output should be like

```
      *  
     * * *  
    * * * * *  
   * * * *  
  * * *  
 * *  
*
```

7. Write a program that displays square, a cube of a number in table form (Use escape sequence)

Output should be like

Number	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64