

## LAB MANUAL 01

---

### INTRODUCTION TO COMPUTER PROGRAMMING

---

#### Lab Objectives:

At the end of this lab students will know about

- Basics of Computer?
- Introduction to Dev-C++ IDE(Integrated Development Environment)
- Simple C++ program in DEV
- Understand program

#### COMPUTER

There could be many definitions of computer, few are

*“It is a combination of hardware and software capable of doing mathematical and logical operations”*

**Or**

*A bit complex one*

*“A computer is a programmable machine. The two principal characteristics of a computer are: It responds to a specific set of instructions in a well-defined manner and it can execute a prerecorded list of instructions (a program).”*

There are two types of memories:

- Primary Memory
- Secondary Memory

## 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://bloodshed-dev-c.en.softonic.com/>

## 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

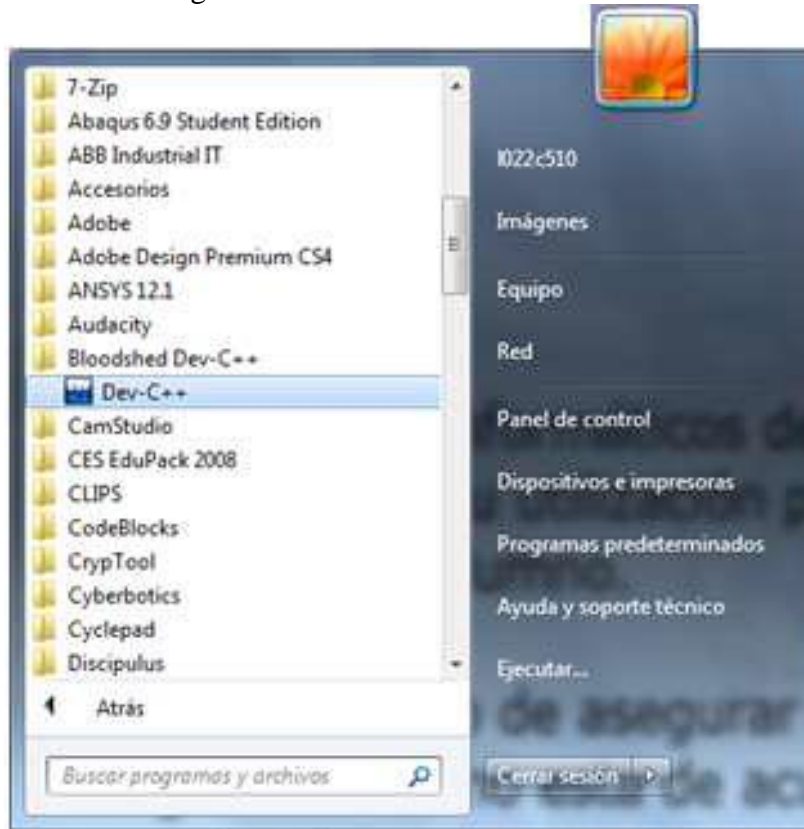


Figure 1. Running Dev-C++ in the computer lab

### b. Create a new source file

File → New → Source File

Or

Ctrl+N

### c. IDE

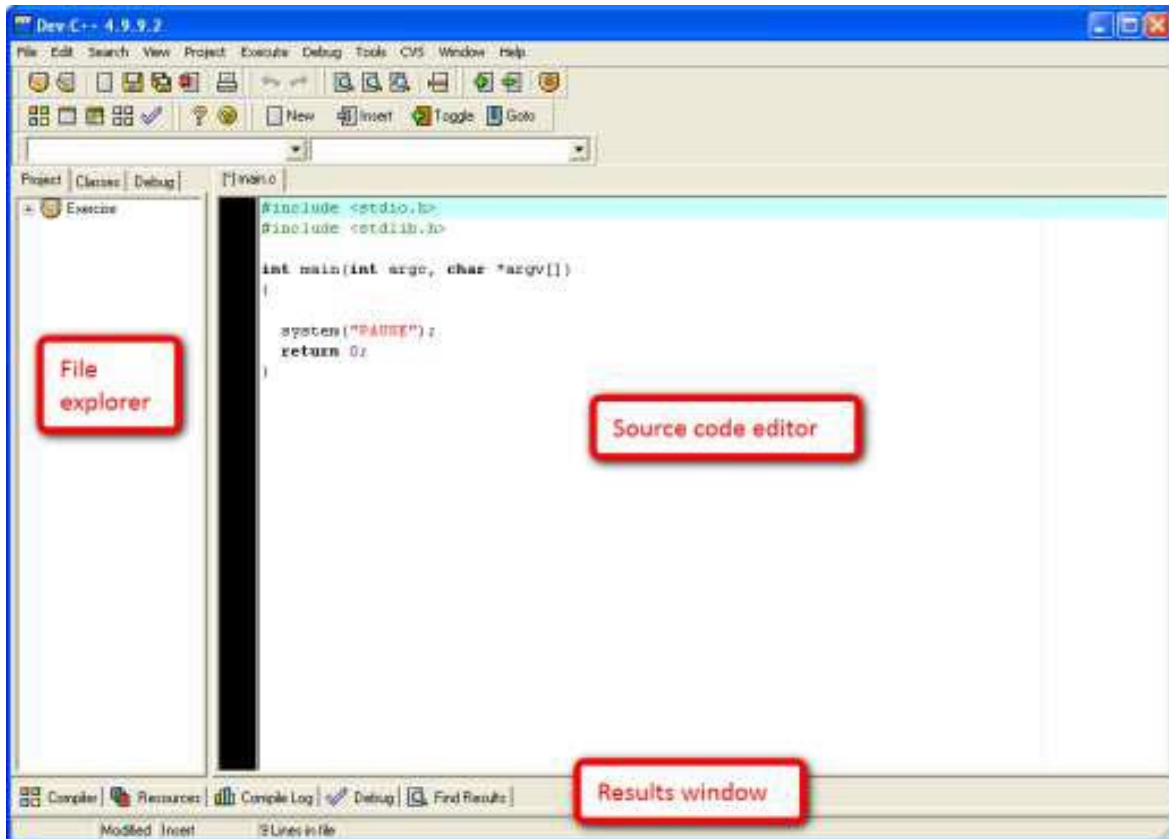
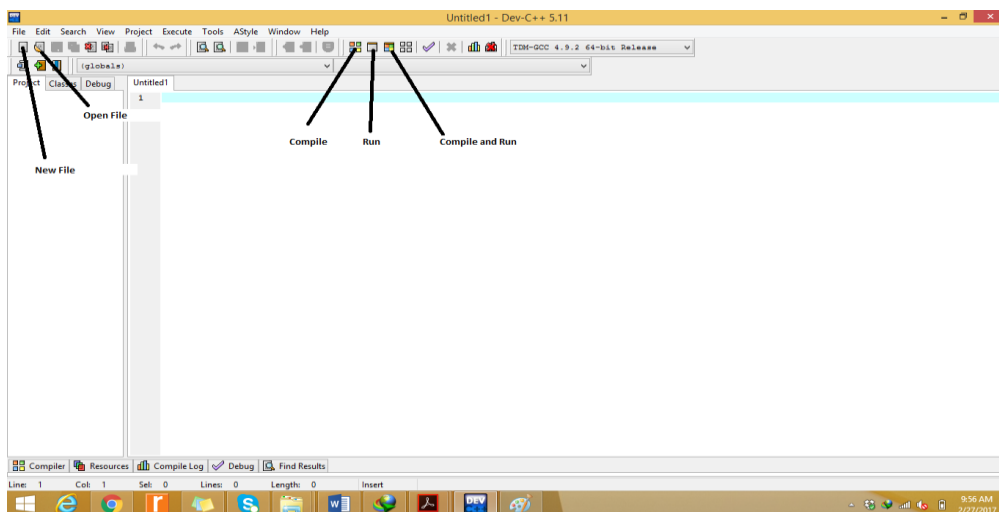


Figure 02. Dev-C++ launches after project creation

- d. Write code and Save file as Program\_name.cpp
- e. For Compiling and Running program



## 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";

    return 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:** The compiler ignores white spaces. However, multiple lines makes the 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!";
```

## **TASKS:**

### **Question #01**

**Print the below patterns?**

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*

Xx

XXXXXXXX

XXXXXXXXXXXX

### **Question #02**

**Make your CV that include your Name, Father's Name, CNIC, Qualification, Semester, CGPA etc.? Print each line separately using “\n” and “endl”? Use comments also.**

### **Question #03**

**Prepare list of available header files and their respective functions(Memorize them).**

### **Question #04**

**Describe phases of compilation.**

### **Question #05**

**Describe different types of computer languages(atleast 8).**