

LAB 1

CSE225L



C++ & OOP Basics

In this lab, we will:

- Familiarize students with the C++ environment and coding style.
- Demonstrate the use of **cout**, **cin**, and other basic commands.
- Show how to create classes and objects, with a focus on **.h** files, **.cpp** files, and how to include them.

Before starting the tasks, ensure you create a **C++ project** in your development environment. The project will contain all the relevant files (source code, header files, etc.) and is necessary for debugging purposes in **Code::Blocks**.

STEPS TO CREATE A PROJECT

1. **File** → **New** → **Project** → **Console Application** → **Go**
2. The "**Console Application**" wizard will appear:
 - Click **Next**.
 - Select **C++** and click **Next**.
 - In the **Project Title** field, enter a name for your project (e.g., lab01).
 - Set the **Folder to create the project in** (e.g., d:\\labs).
 - Accept default settings for the rest and click **Next**.
 - In the **Compiler** field, leave it as **GNU GCC Compiler** and click **Finish**.

A new project folder (e.g., d:\\labs\\lab01) will be created with the configuration file lab01.cbp.

3. Under the **Management** pane → Choose **Projects** tab → Expand the **lab01** node → Expand the **Sources** node → Double-click main.cpp to open the default "Hello, World!" program.
 4. To **build** the program, go to the **Build** menu → Click **Build**.
 5. To **run** the program, go to the **Build** menu → Click **Run**.
-

C++ Basics

TASKS:

1. Take two integer inputs from the keyboard and print their sum.
 2. Take an integer input from the keyboard and calculate its factorial.
 3. Take two integer inputs from the keyboard and ask the user which operation to perform: '+', '-', or '*'. Perform the appropriate operation and display the result. (Hint: you can use a switch/case statement.)
 4. Write a function **isPrime(int n)** that checks if 'n' is a prime number. Use this function to print all the prime numbers between 300 and 500.
-

OOP Basics

TASKS:

1. Create a class called **Box** that has height, width, and depth. The class should include a method **volume()** that calculates and displays the volume of the box.
2. Create a driver file (main.cpp) where you will instantiate multiple **Box** objects with arbitrary dimensions and display their volumes.