# DATA STRUCTURES

# Fall 2023

# LAB 01

**Learning Outcomes**

**In this lab you are expected to learn the following:**

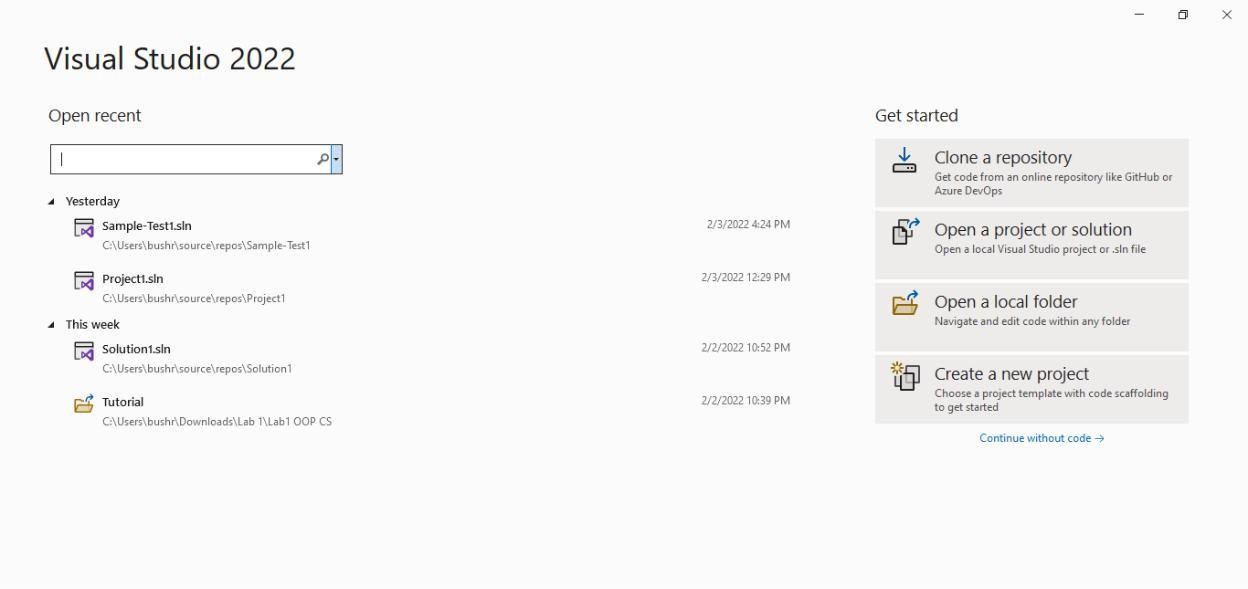
**An overview of**

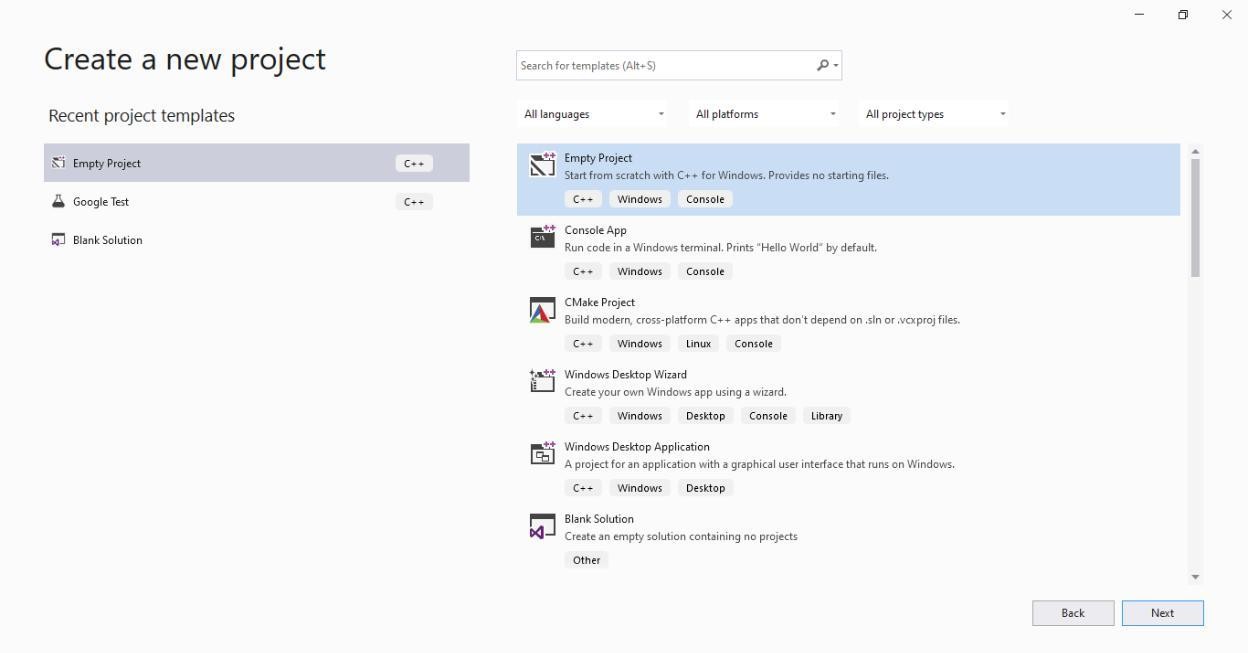
* **OBJECT ORIENTED PROGRAMMING**
* **CLASSES**
* **G Test**

# Getting started with Visual Studio

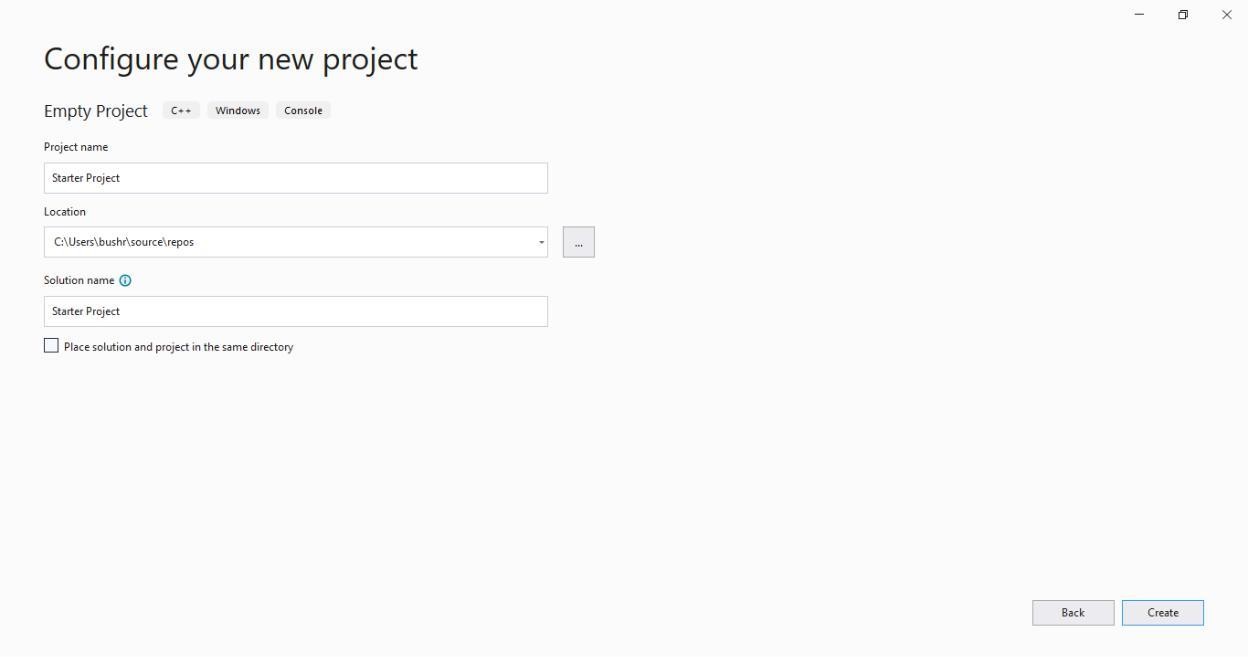
The usual starting point for a C++ programmer is a "Hello, world!" application that runs on the command line. That's what you'll create first in Visual Studio, and then we'll move on to something more challenging



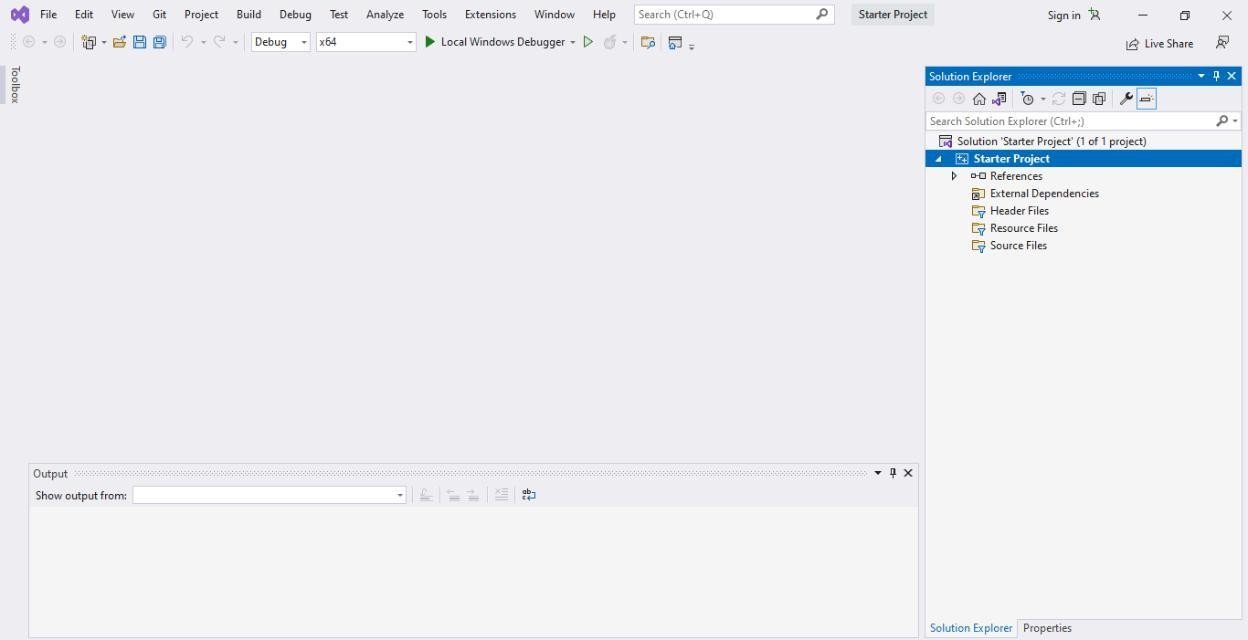
1. Choose Create a new project to get started.
2. In the list of project templates, choose Empty Project, then choose Next.



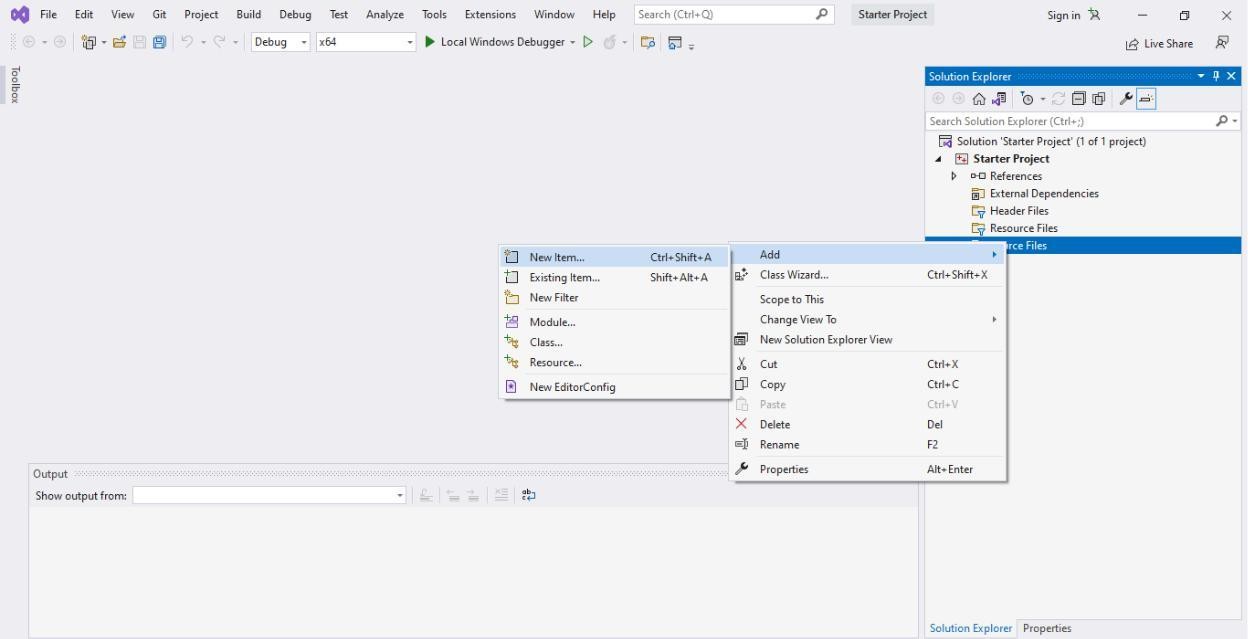
1. In the Configure your new project dialog box, select the Project Name edit box, name your new project Starter Project, then choose to Create.



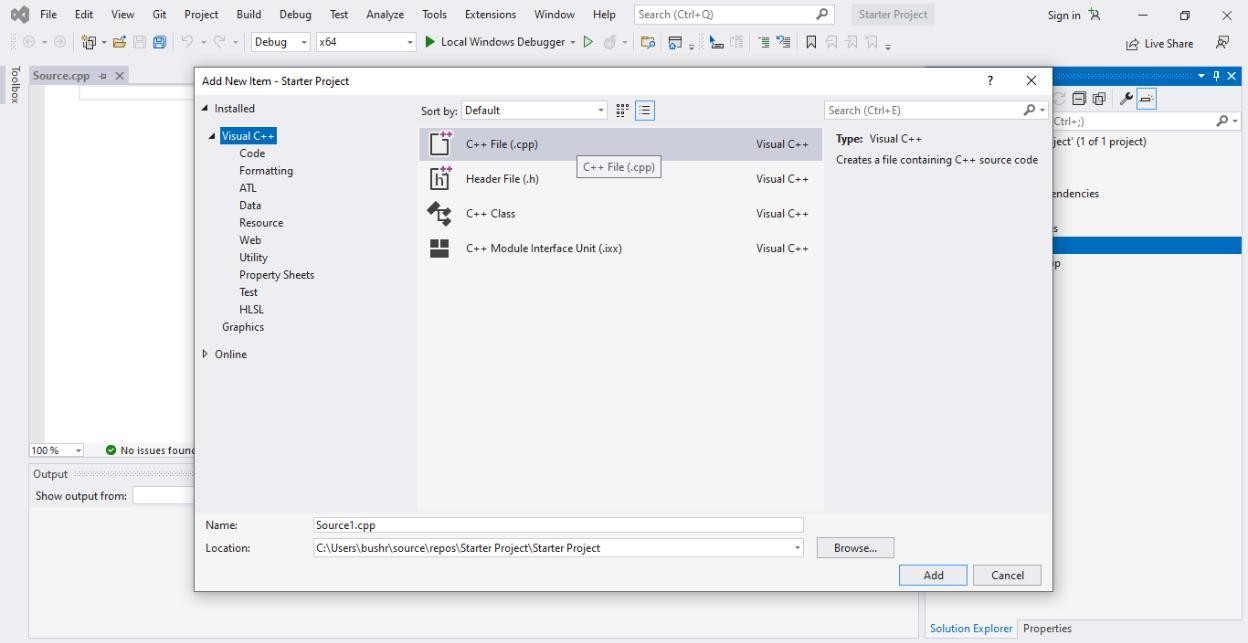
1. An empty C++ Windows console application gets created. Console applications use a Windows console window to display output and accept user input. In Visual Studio, an empty project will be created.



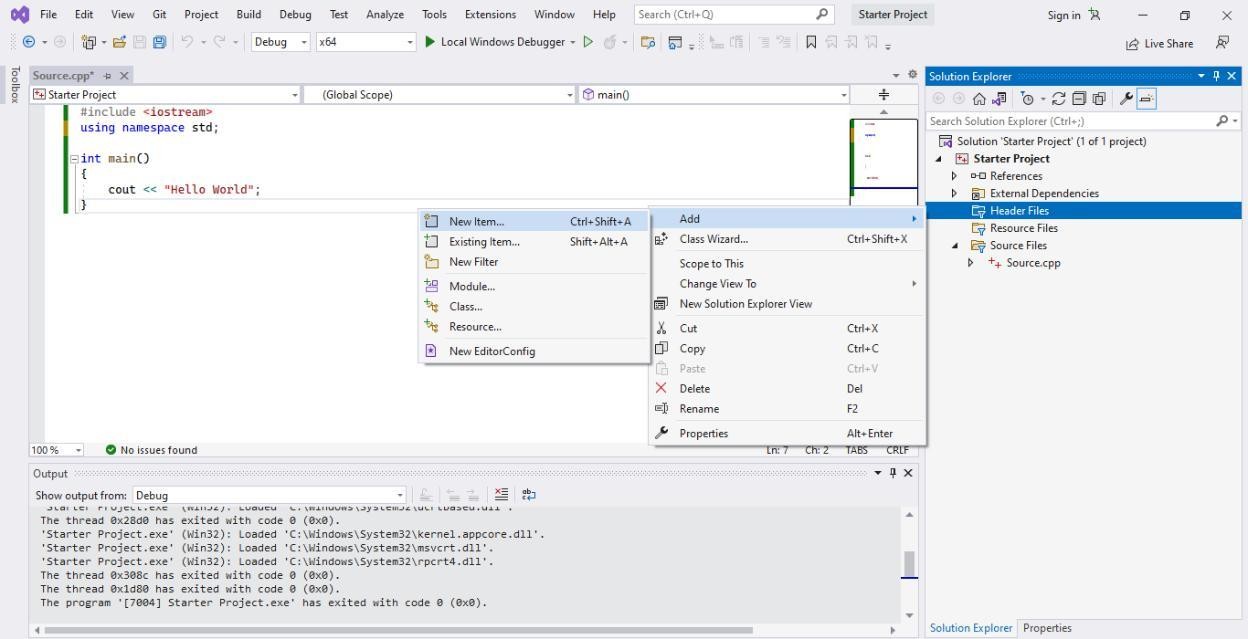
1. Right-click on Source Files and add a new item.



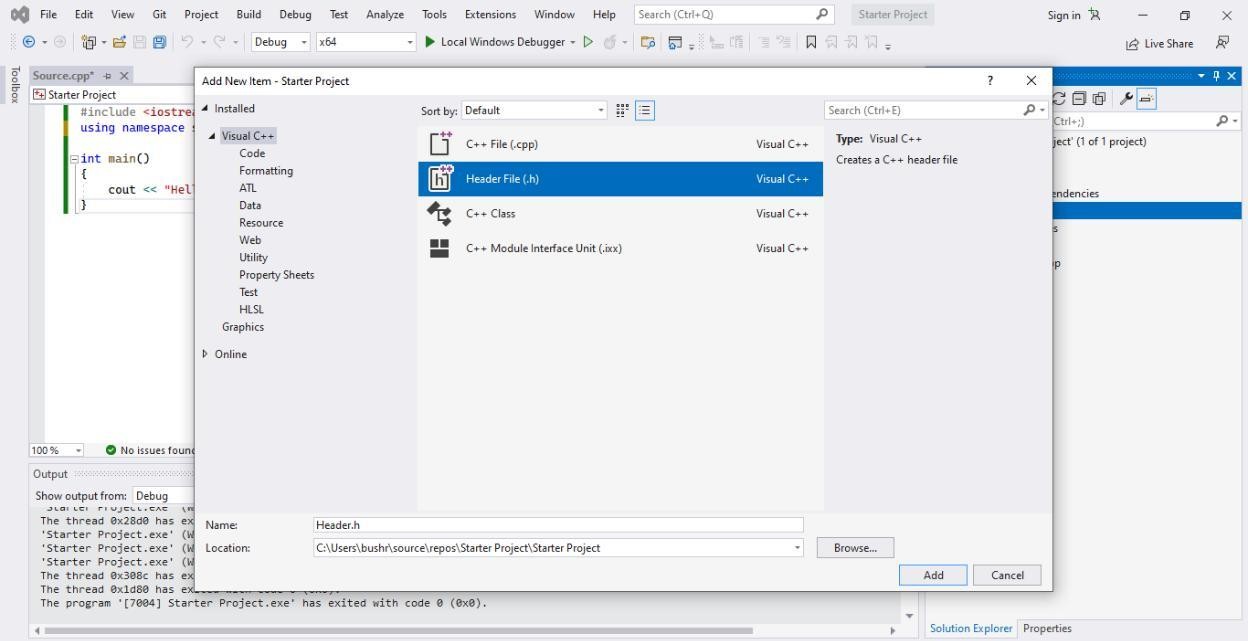
1. Create .cpp file and click Add



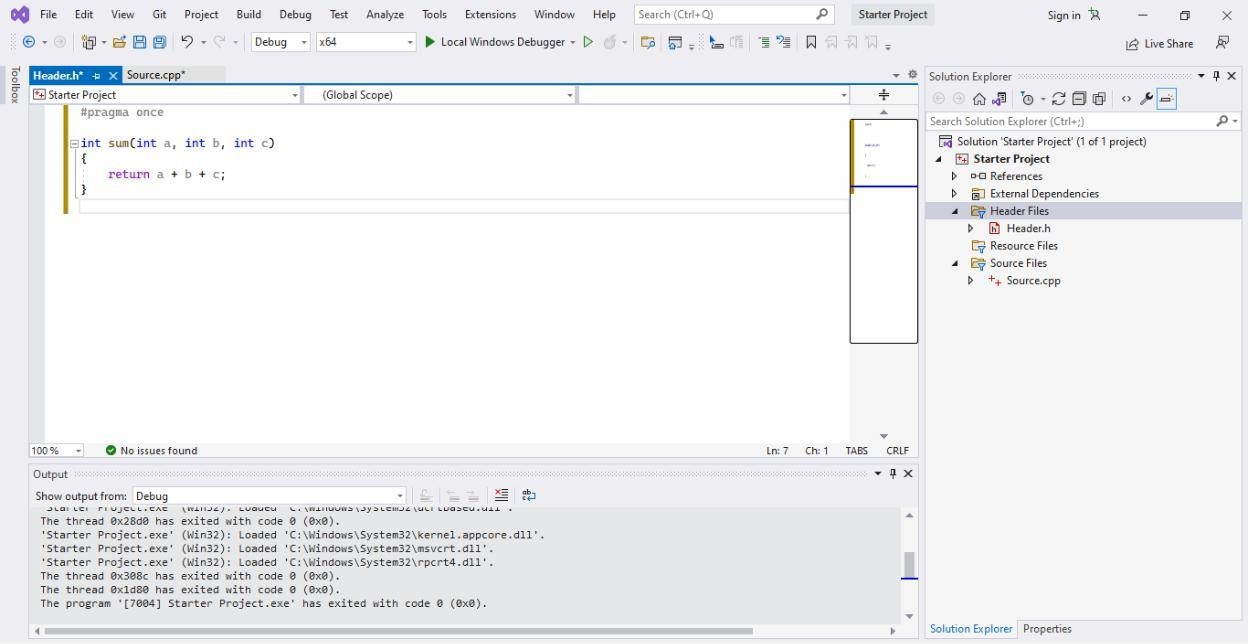
1. Right-click on Header Files and add a new item.



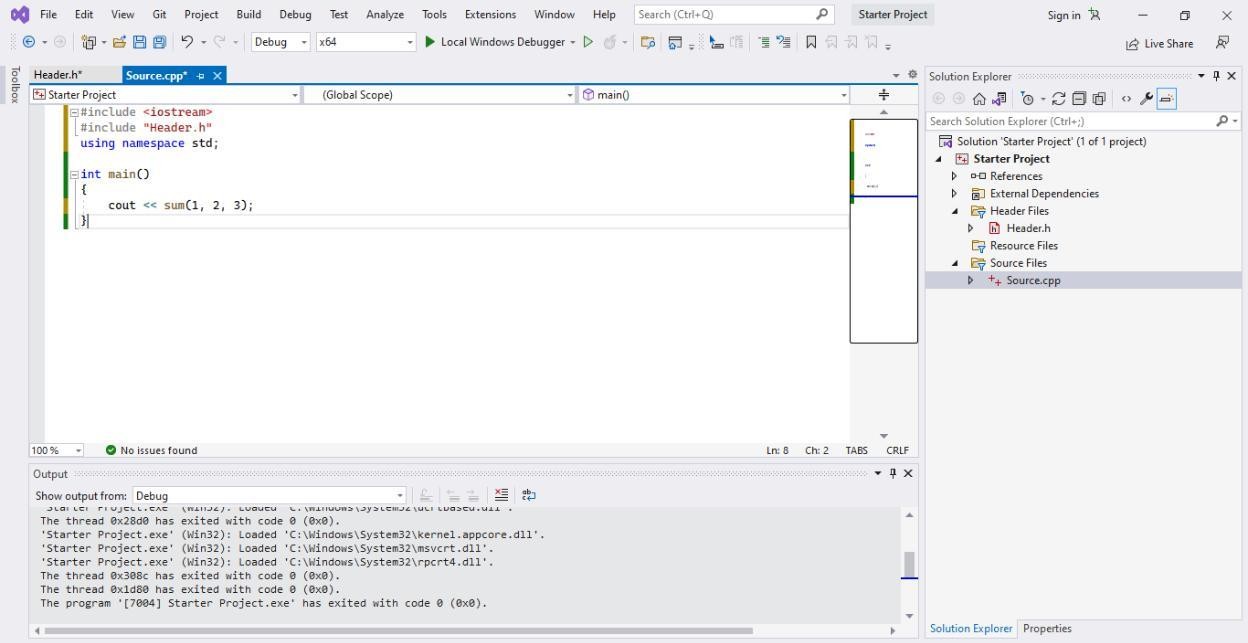
1. Select header file and click add



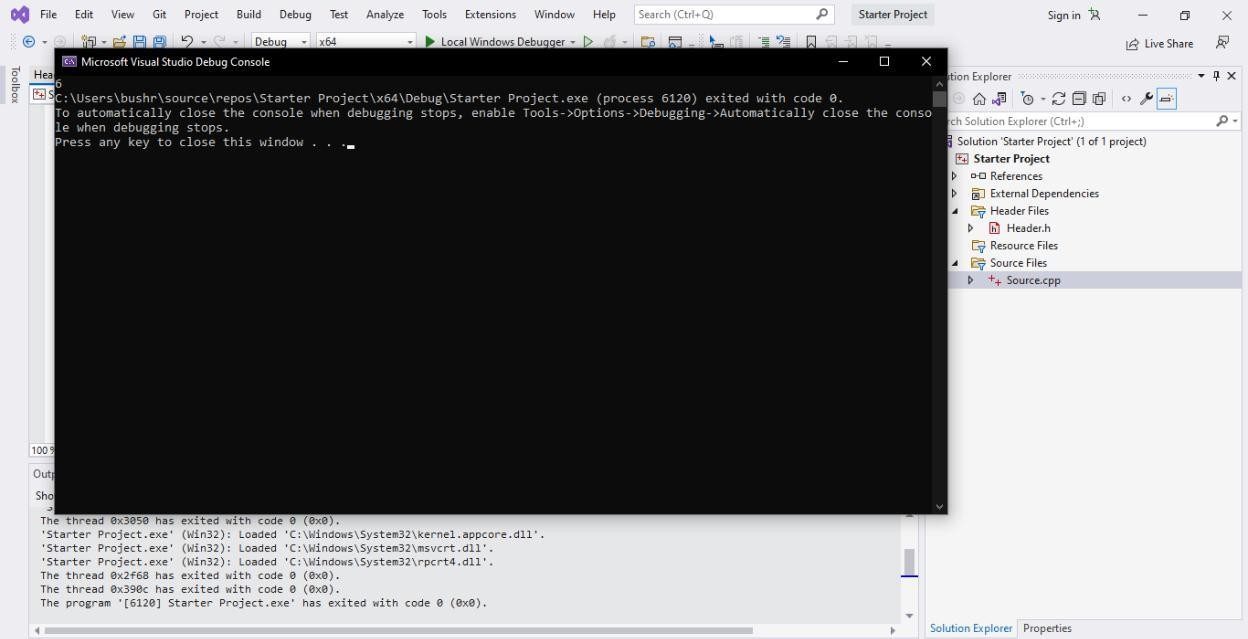
1. This is where you are going to write all your functions. Write sum() function in header file



1. You can use the functions written in the header file by including the file in Source.cpp.



1. Compile and run the code using the Local Windows Debugger option from the quick access toolbar followed by the green play button.



# Getting started with Gtest

Google test is a framework for writing C++ unit tests.

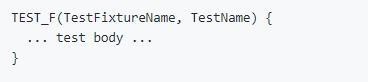
When using googletest, you start by writing *assertions*, which are statements that check whether a condition is true.

***Tests*** use assertions to verify the tested code’s behavior. If a test crashes or has a failed assertion, then it *fails*; otherwise it *succeeds*.

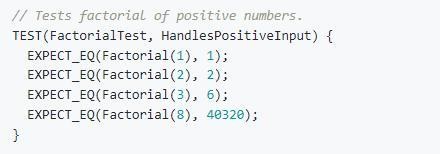
A ***test suite*** contains one or many tests. You should group your tests into test suites that reflect the structure of the tested code.

A ***test program*** can contain multiple test suites.

This is how a test suite looks like,



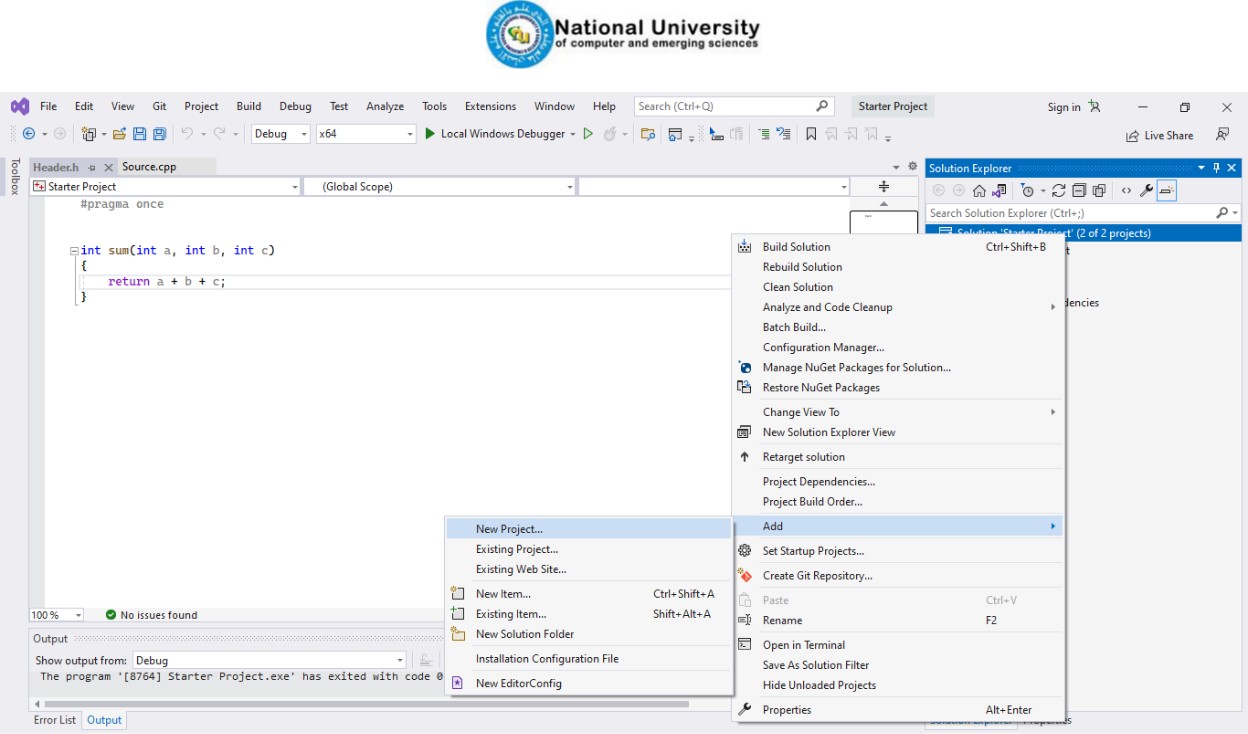
Example



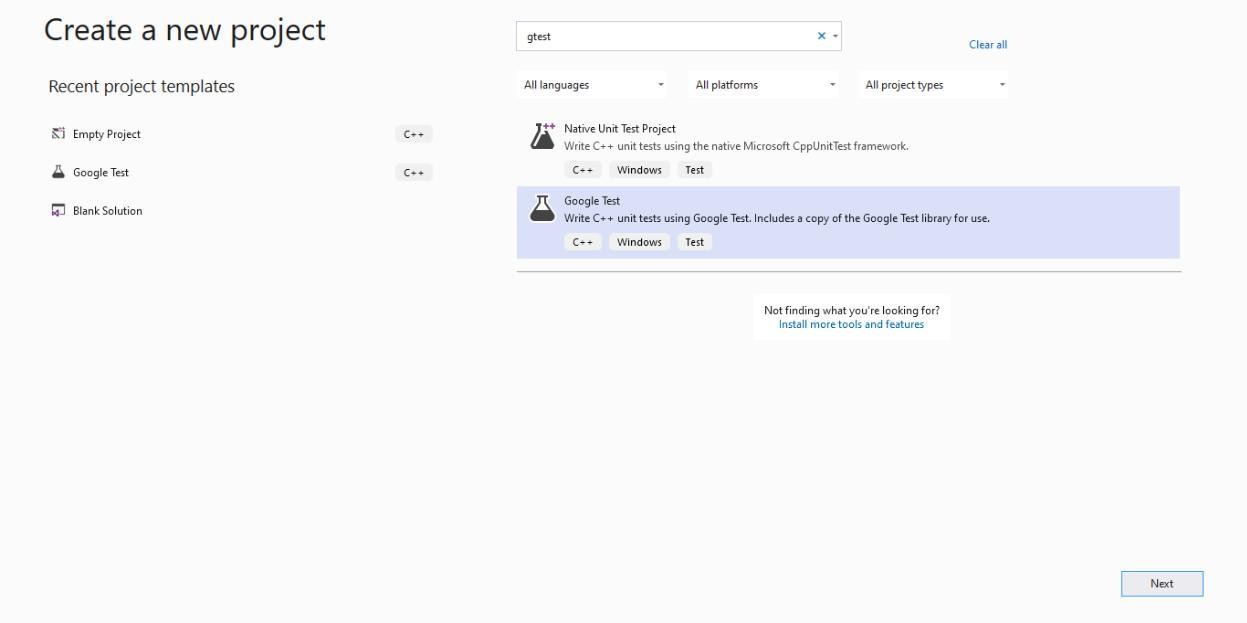
# Visual Studio

In Visual Studio 2017 and later, Google Test is integrated into the Visual Studio IDE as a default component of the Desktop Development with C++ workload.

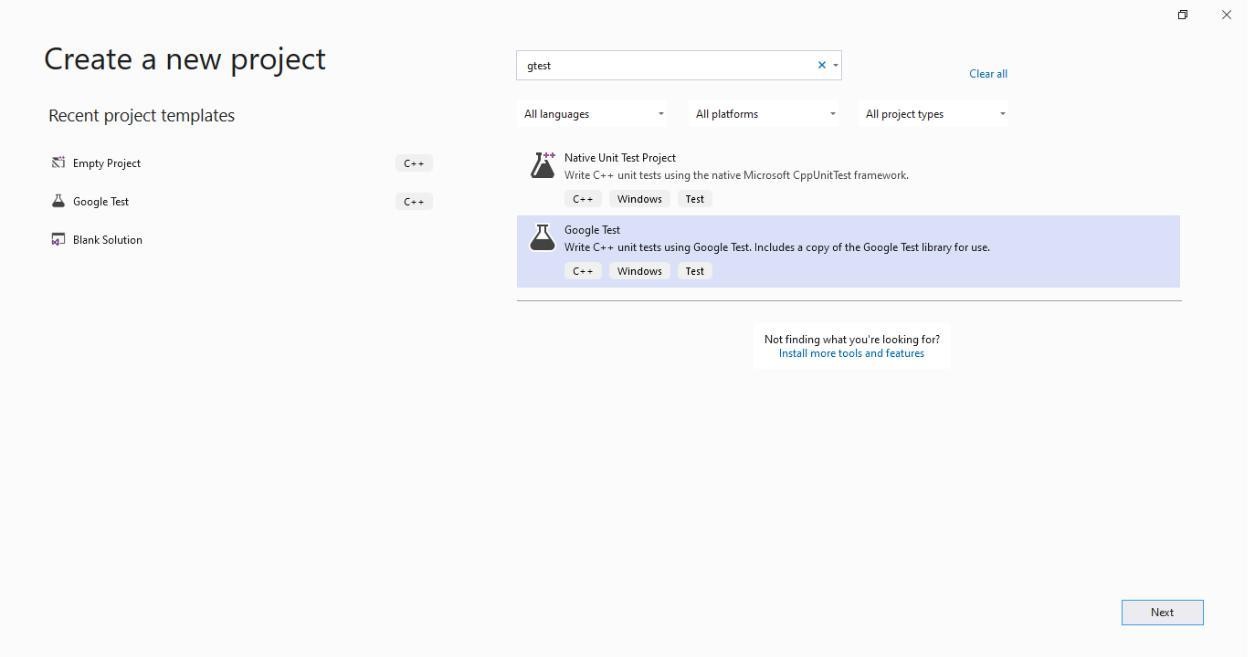
1. In Solution Explorer, right-click on the solution node and choose Add > New Project. Add a Google Test project in Visual Studio. Click on File and create a new project



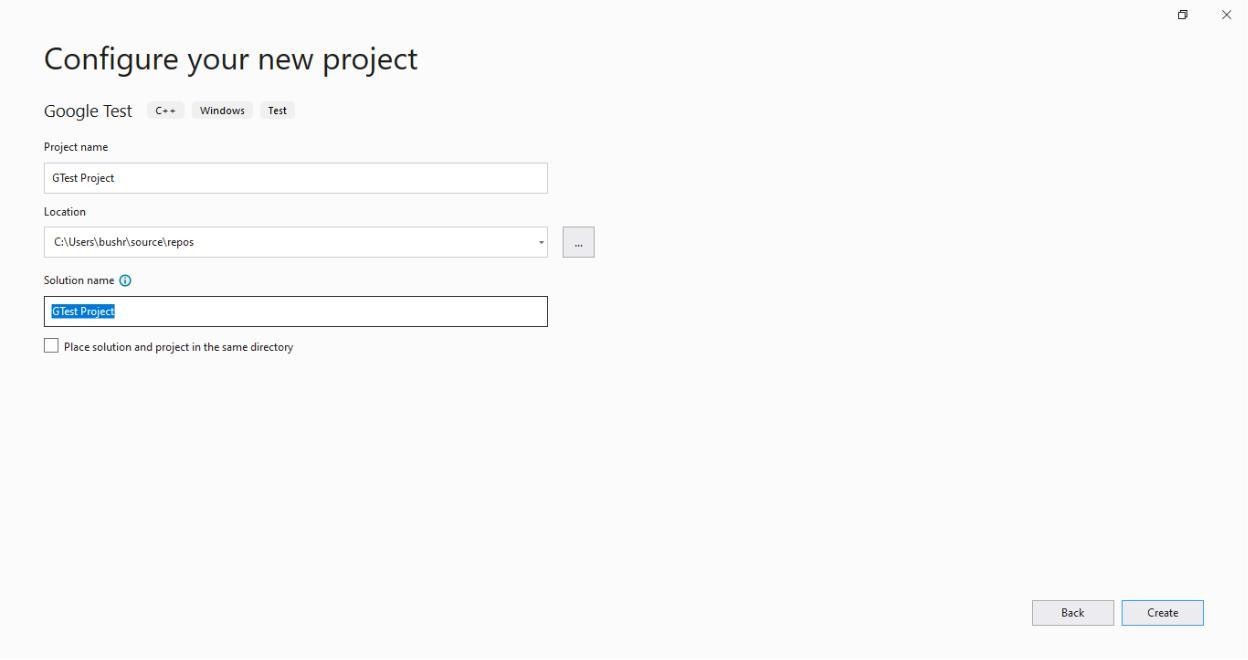
1. Search for google test and click next



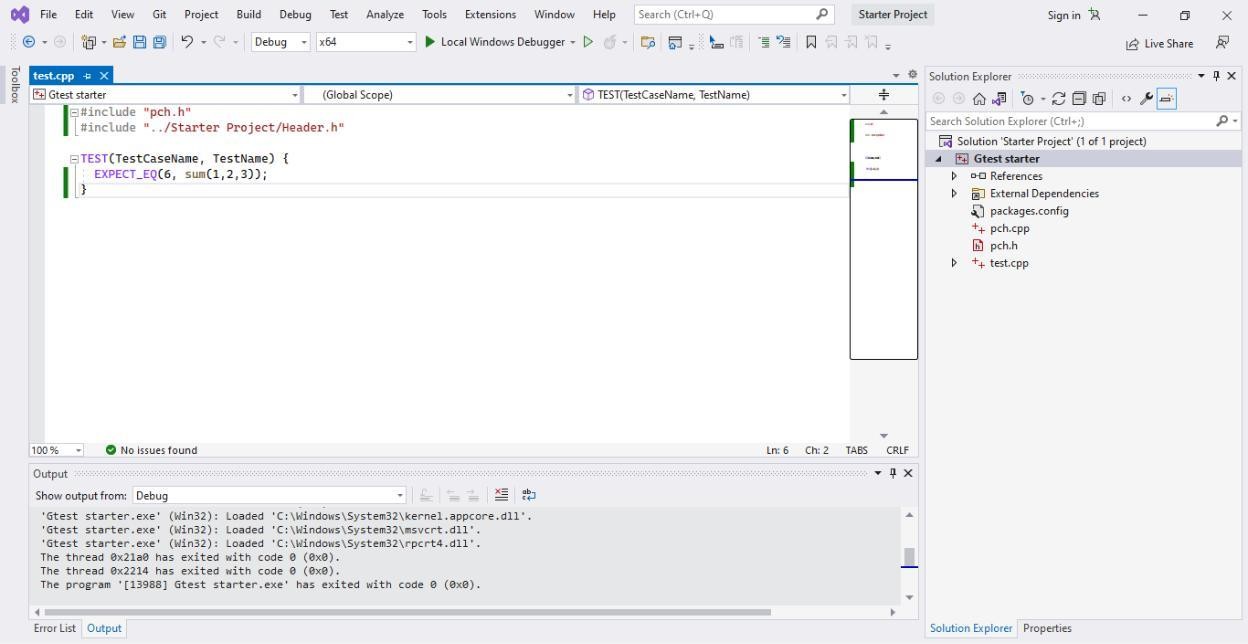
1. Rename the project and click create



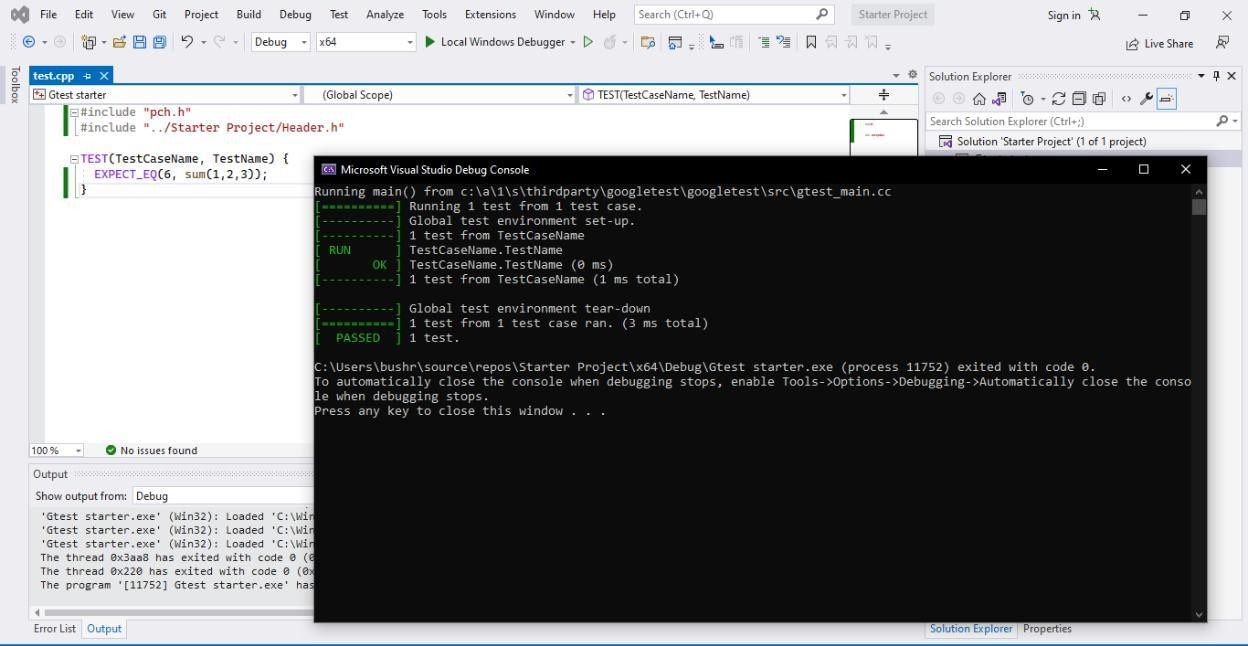
1. Configure your project and click create



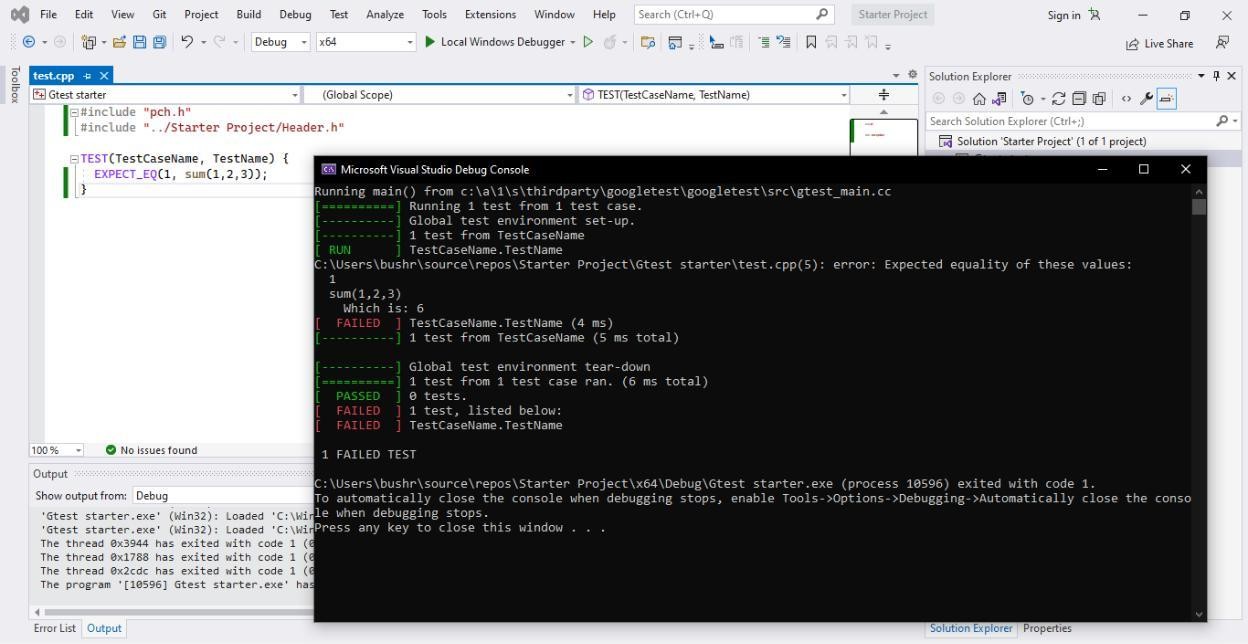
1. You are now ready to write and run Google Tests. Include the header file you want to test and write the test case



1. Compile and run the code using the Local Windows Debugger option from the quick access toolbar followed by the green play button
   1. Passed Test Case



* 1. Failed test case



**Lab Task**

1. **OBJECT ORIENTED PROGRAMMING**

Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

Object-oriented programming has several advantages over procedural programming:

OOP is faster and easier to execute

OOP provides a clear structure for the programs

OOP helps to keep the C++ code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug

OOP makes it possible to create full reusable applications with less code and shorter development time

1. **CLASSES**

A Class is a user defined data-type which has data members and member functions.

Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions defines the properties and behavior of the objects in a Class.

In the above example of class Car, the data member will be speed limit, mileage etc and member functions can be apply brakes, increase speed etc.

An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

1. **POINTER TO OBJECTS**

A pointer to a C++ class is done exactly the same way as a pointer to a structure and to access members of a pointer to a class you use the member access operator -> operator, just as you do with pointers to structures. Also as with all pointers, you must initialize the pointer before using it

## 

## 

## Task 1

Write a class **Patient** that has following data members Name (string), type e.g in – patient, outpatient (string), Number of days (int). Write another class **Hospital** that has following data members patient Ob.(patient), meditation-charges (int), daily rate (double) and hospital charges(int).

Make overloaded constructors in both classes.

First, the program should ask if the patient was **admitted** as an **in-patient** or an **out-patient**. If the patient was an **in-patient**, the following data should be entered:

**1. The number of days spent in the hospital**

**2. The daily rate**

**3. Hospital medication charges**

**4. Charges for hospital services (lab tests, etc.)**

The program should ask for the following data if the patient was an **out-patient**:

**1. Charges for hospital services (lab tests, etc.)**

**2. Hospital medication charges**

The program should use **two overloaded functions in hospital class** to **calculate the total charges**. **One** of the functions should **accept arguments** for the **in-patient data (daily rate \* hospital charges \* meditation rate \*)**, while the other function **accepts arguments** for **out-patient** information. Both functions should **return the total charges**. Write your **main** function and test the functionality of your application.

**Submission Instructions**

1. Save all .cpp files with your roll no and task number e.g. i22XXXX\_Problem01.cpp
2. Now create a new folder with name ROLLNO\_LAB\_01 e.g. i22XXXX\_LAB\_02
3. Move all of your .cpp files to this newly created directory and compress it into .zip file
4. Now you have to submit this zipped file on Google Classroom.