National University of computer and emerging sciences

OBJECT-ORIENTED PROGRAMMING

BSCS Spring 2024

Lab Manual 01

Learning Outcomes

In this lab you are expected to learn the following:

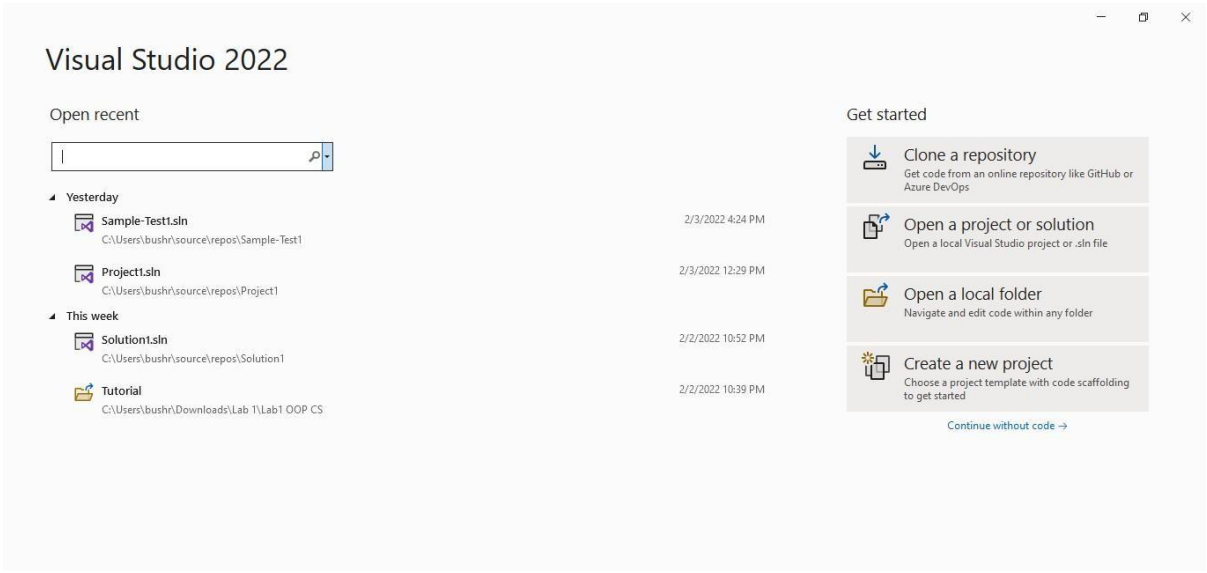
- Debugging
- G-Testing

National University of computer and emerging sciences

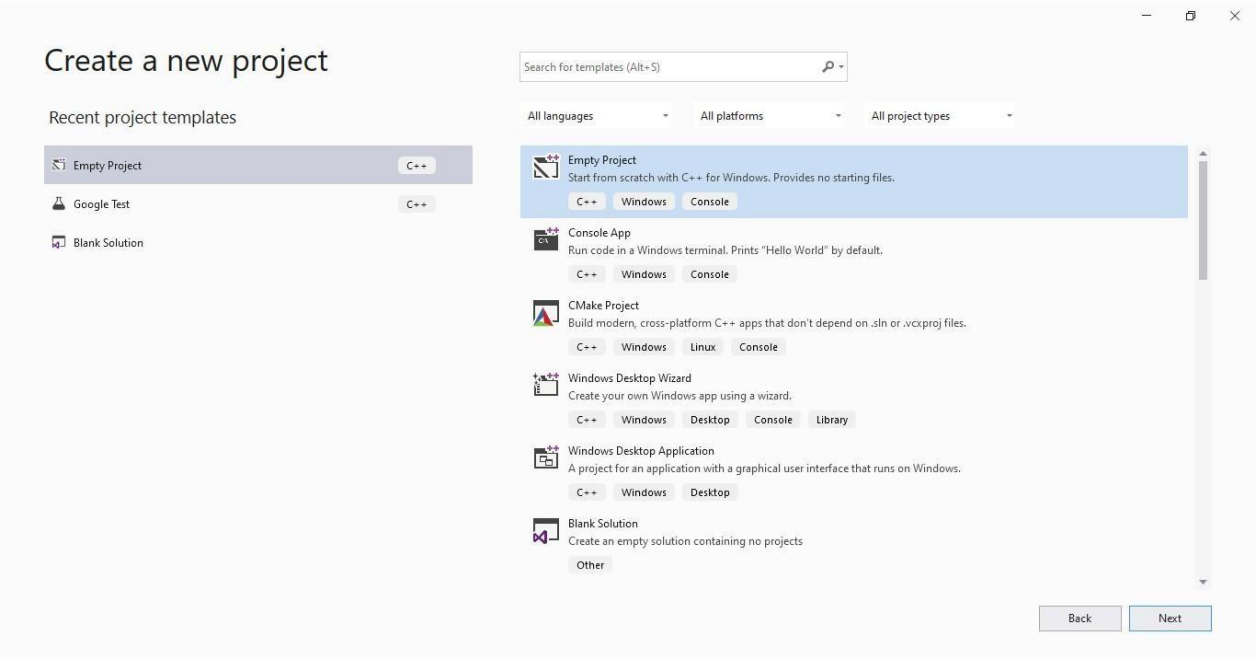
Getting started with Visual Studio

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

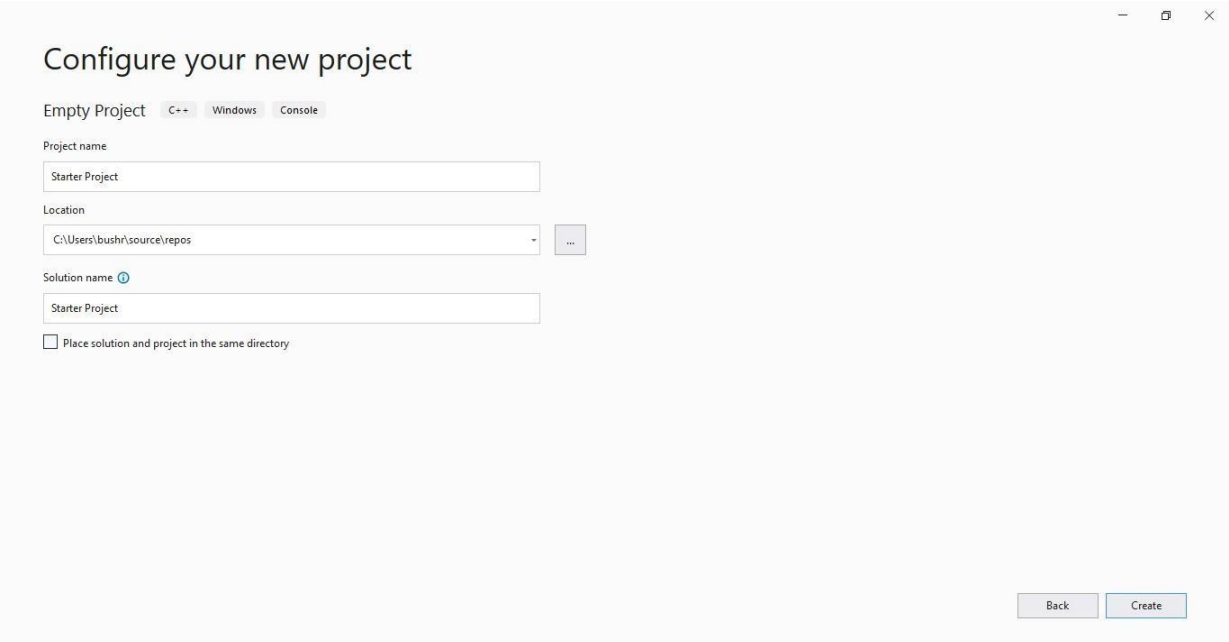
1. Choose Create a new project to get started.



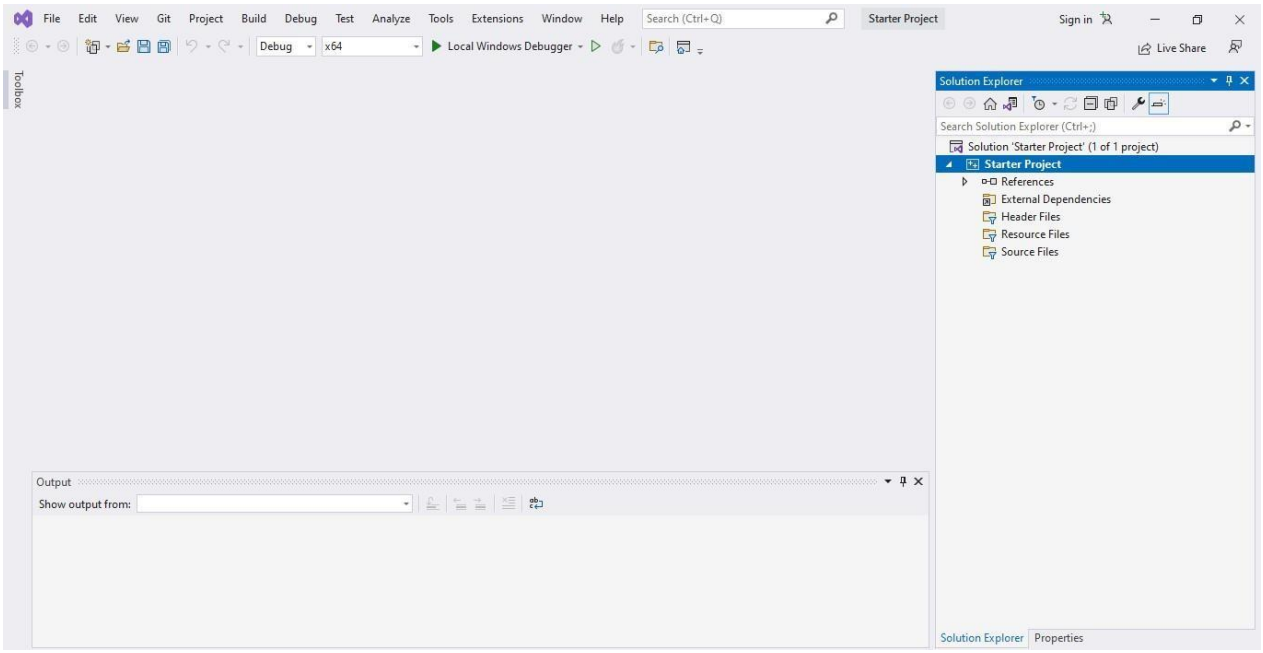
2. In the list of project templates, choose Empty Project, then choose Next.



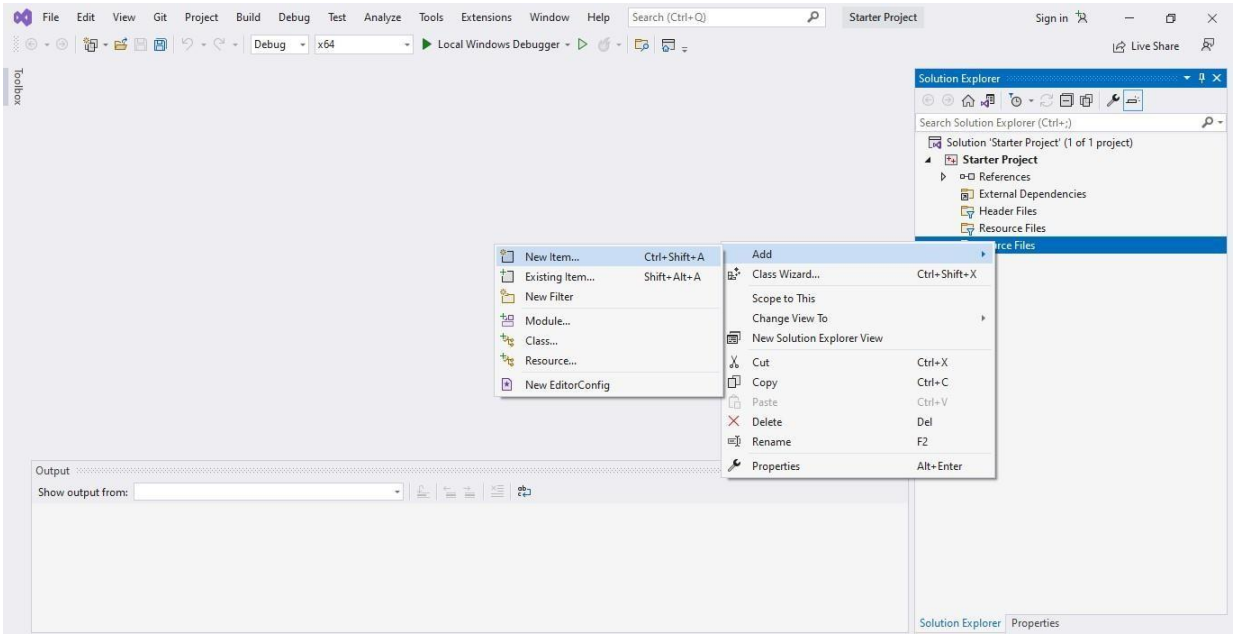
3. In the Configure your new project dialog box, select the Project Name edit box, name your new project Starter Project, then choose to Create.



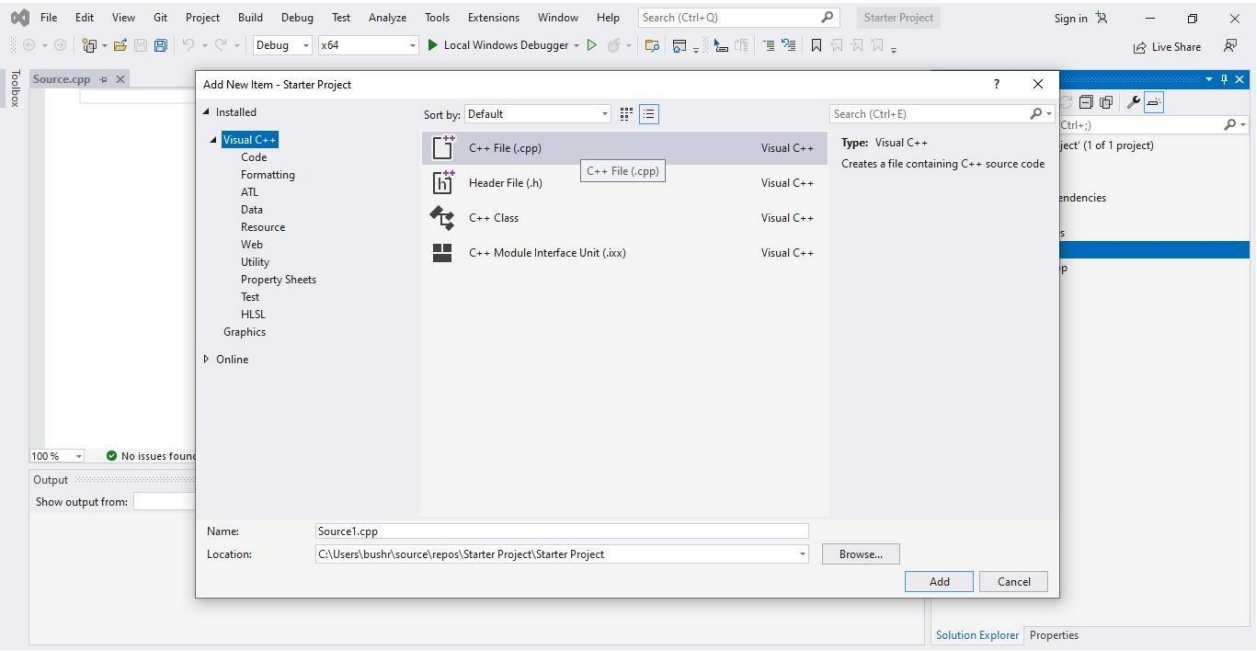
4. An empty C++ Windows console application gets created. Console applications use a Windows console window to display output and accept user input. In Visual Studio, an empty project will be created.



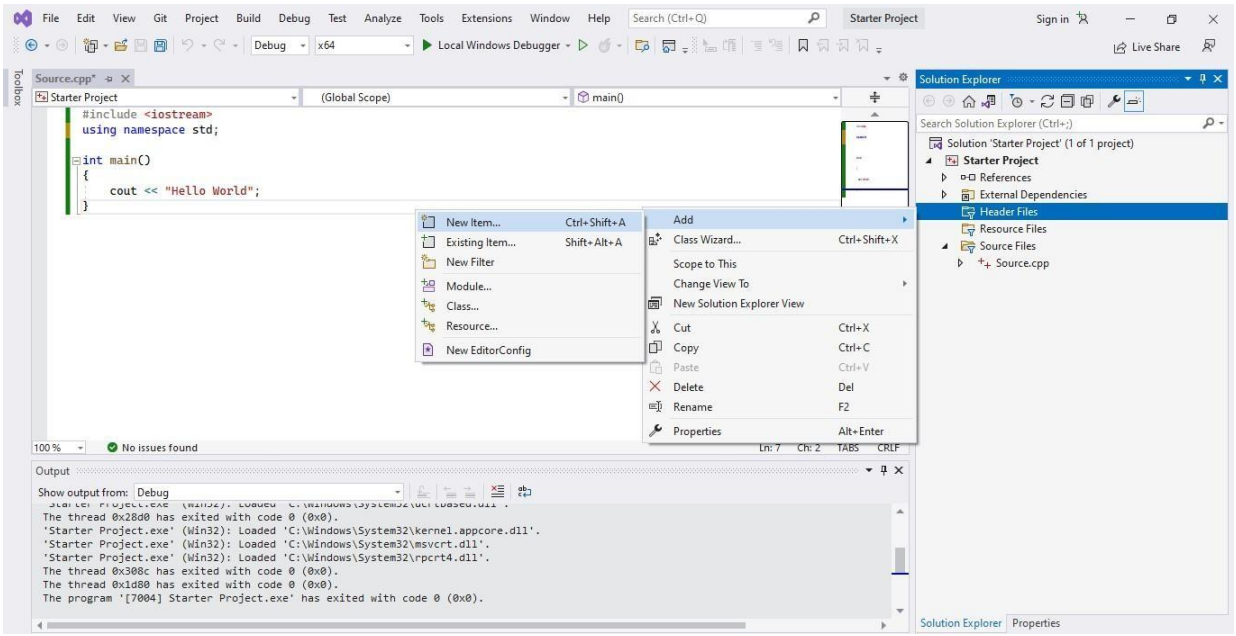
5. Right-click on Source Files and add a new item.



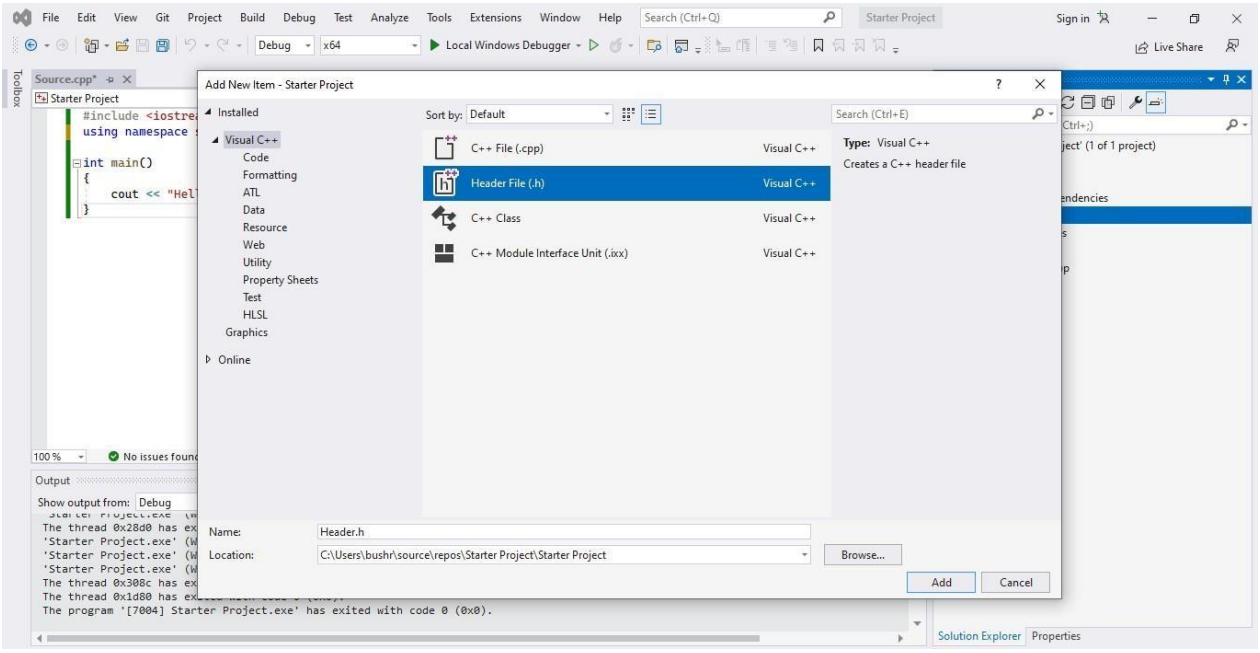
6. Create .cpp file and click Add



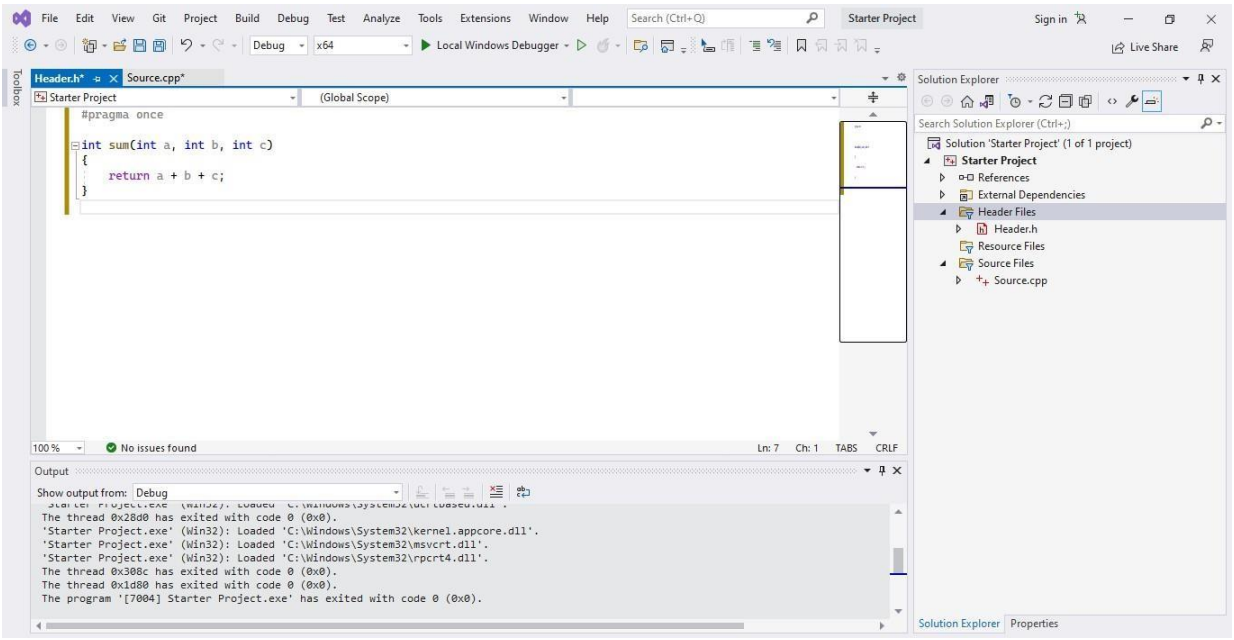
7. Right-click on Header Files and add a new item.



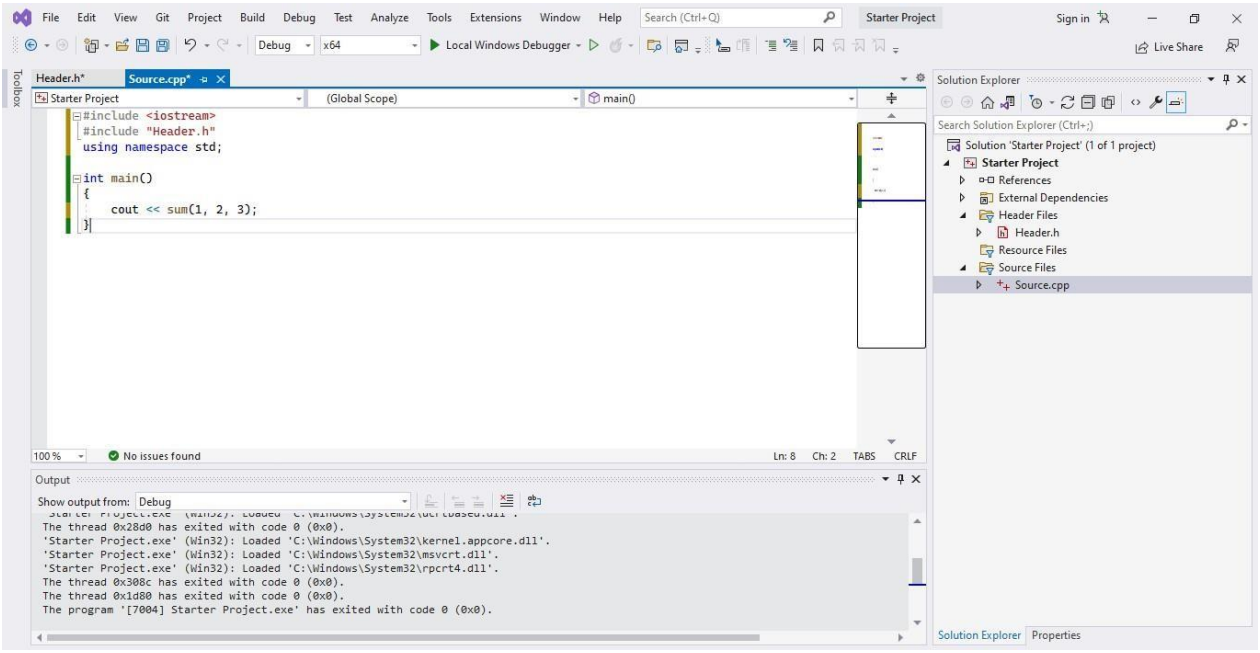
8. Select header file and click add



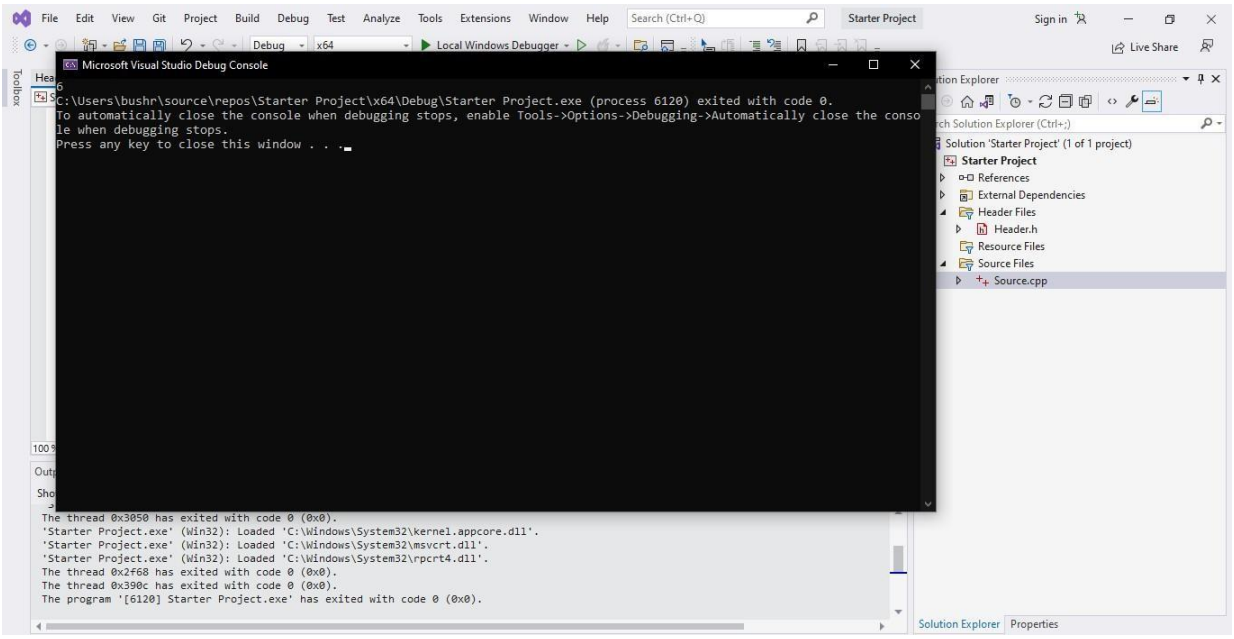
9. This is where you are going to write all your functions. Write sum() function in header file



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



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



National University of computer and emerging sciences

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,

```
TEST_F(TestFixtureName, TestName) {
  ... test body ...
}
```

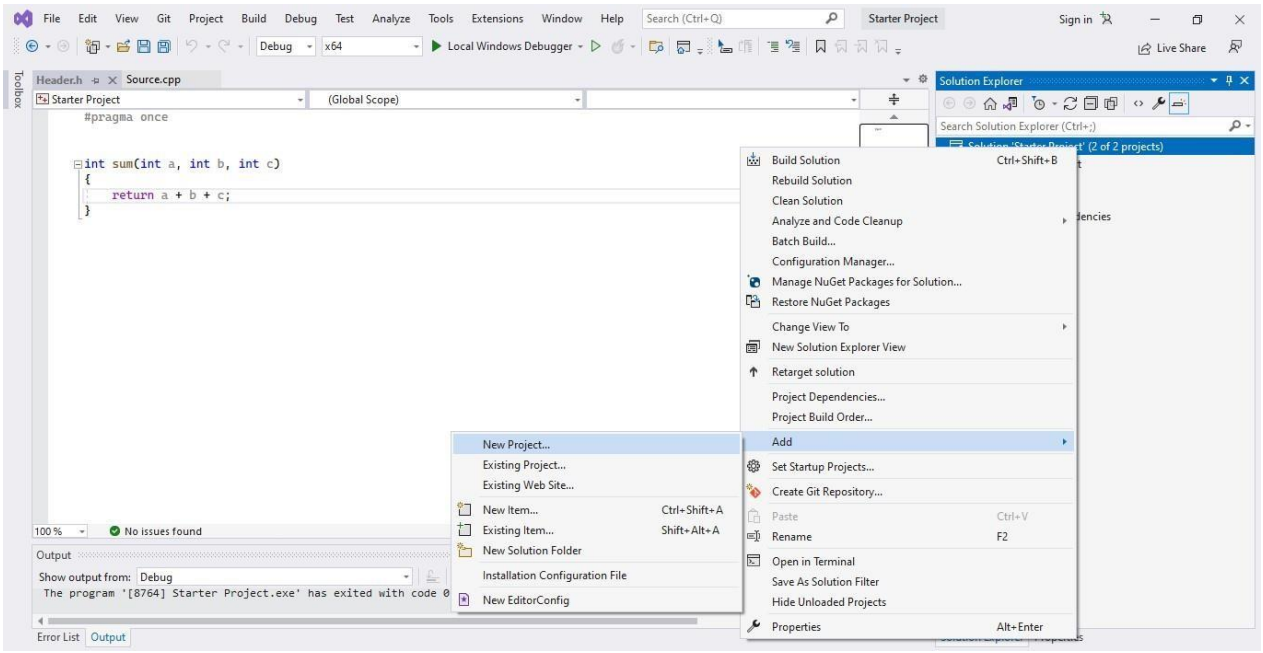
Example

```
// Tests factorial of positive numbers.
TEST(FactorialTest, HandlesPositiveInput) {
  EXPECT_EQ(Factorial(1), 1);
  EXPECT_EQ(Factorial(2), 2);
  EXPECT_EQ(Factorial(3), 6);
  EXPECT_EQ(Factorial(8), 40320);
}
```

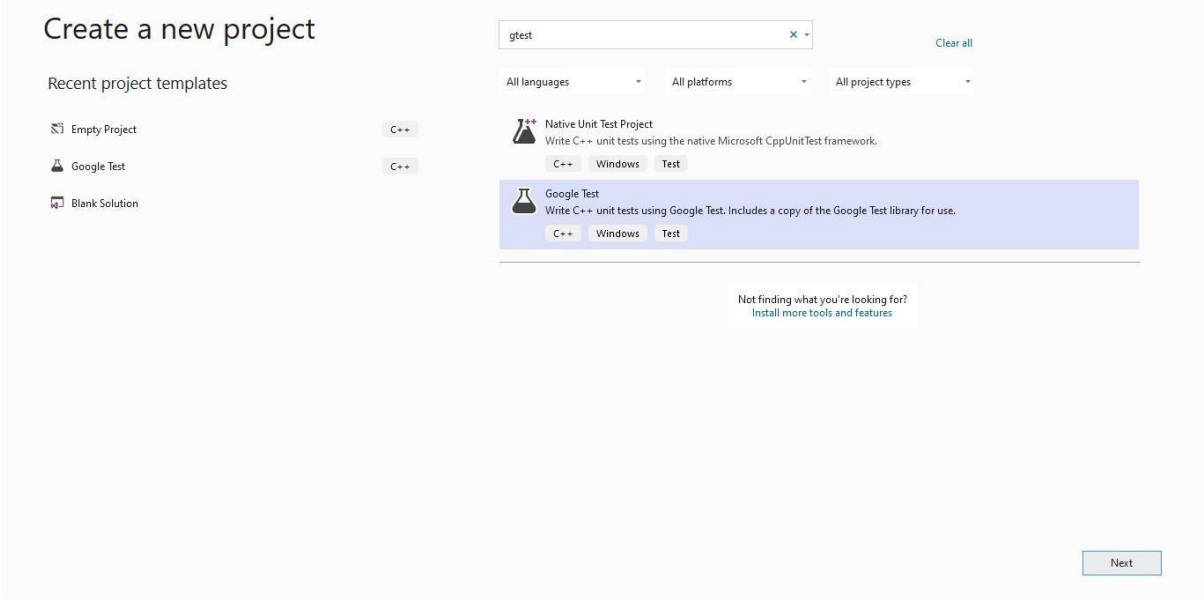
Visual Studio

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.

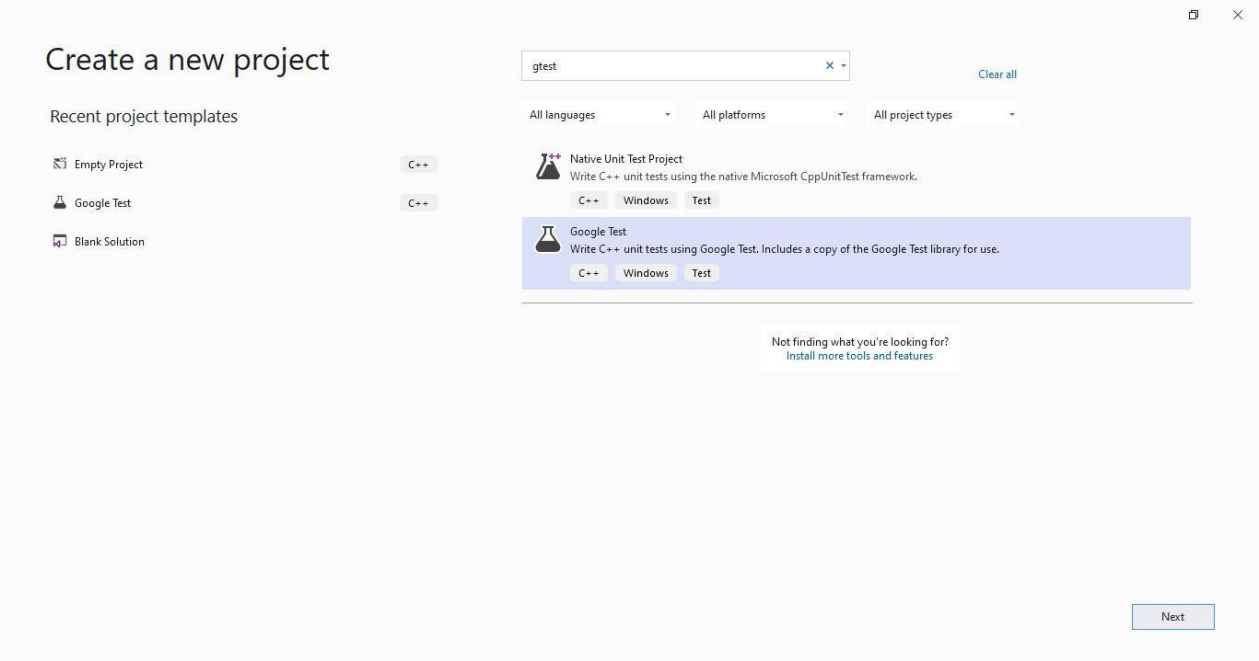
1. In Solution Explorer, right-click on the solution node and choose Add > New Project. Add a Google Test project in Visual Studio. Click on File and create a new project



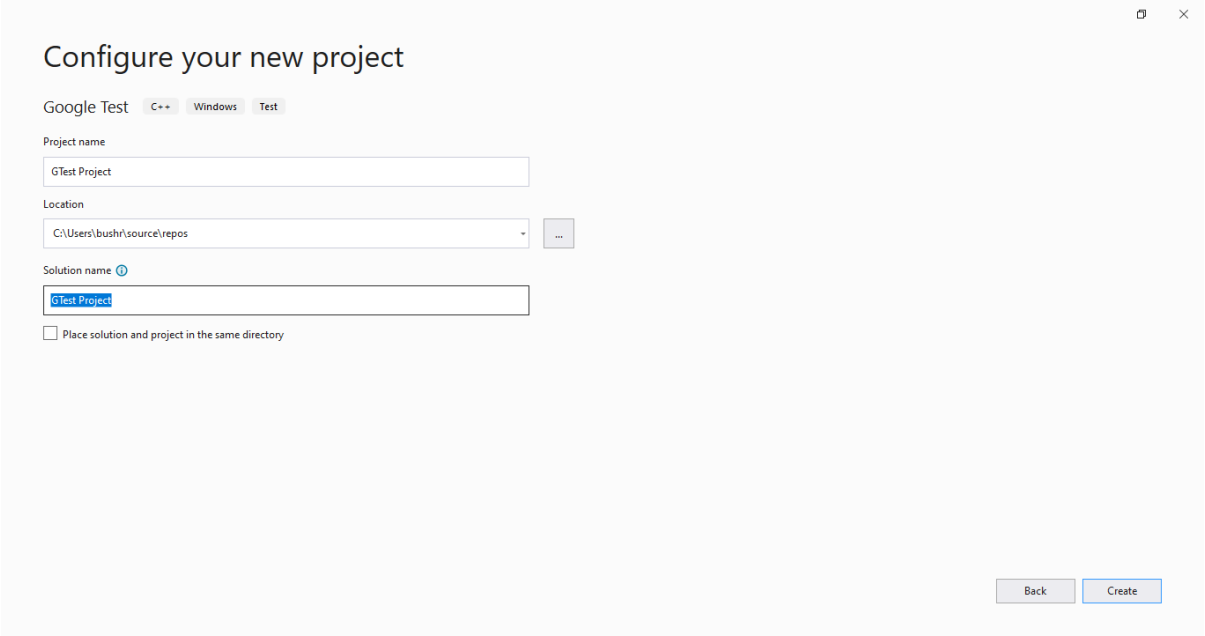
2. Search for google test and click next



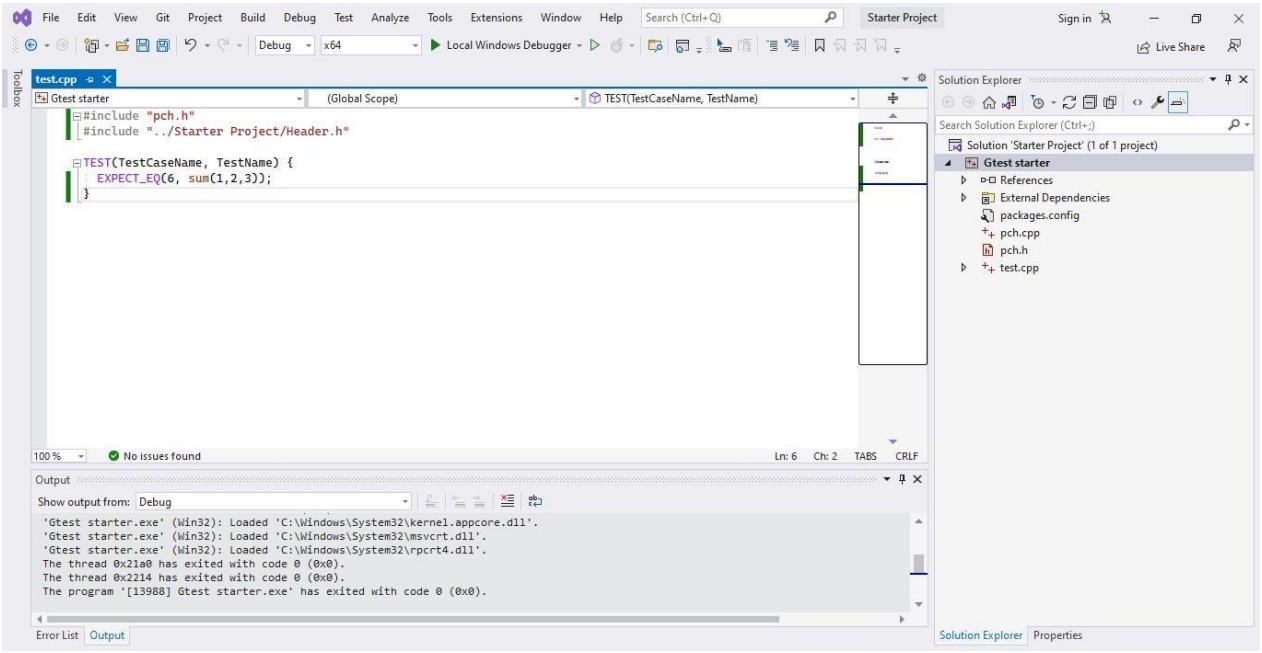
3. Rename the project and click create



4. Configure your project and click create

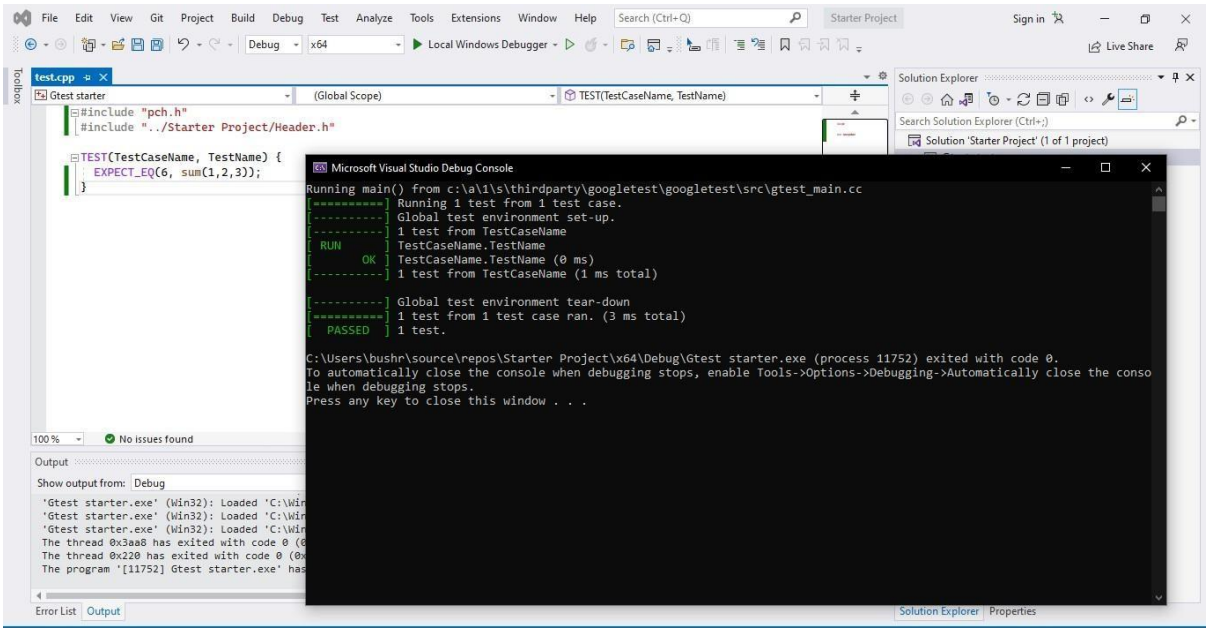


5. You are now ready to write and run Google Tests. Include the header file you want to test and write the test case

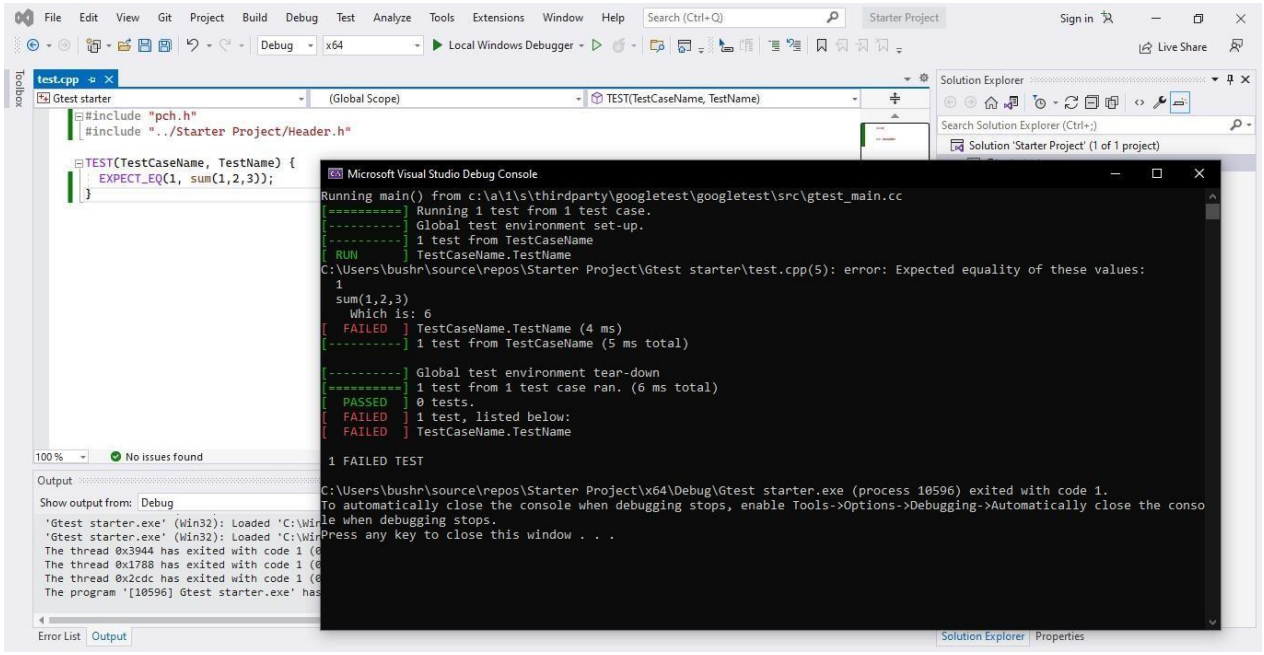


6. Compile and run the code using the Local Windows Debugger option from the quick access toolbar followed by the green play button

    a. Passed Test Case



b. Failed test case

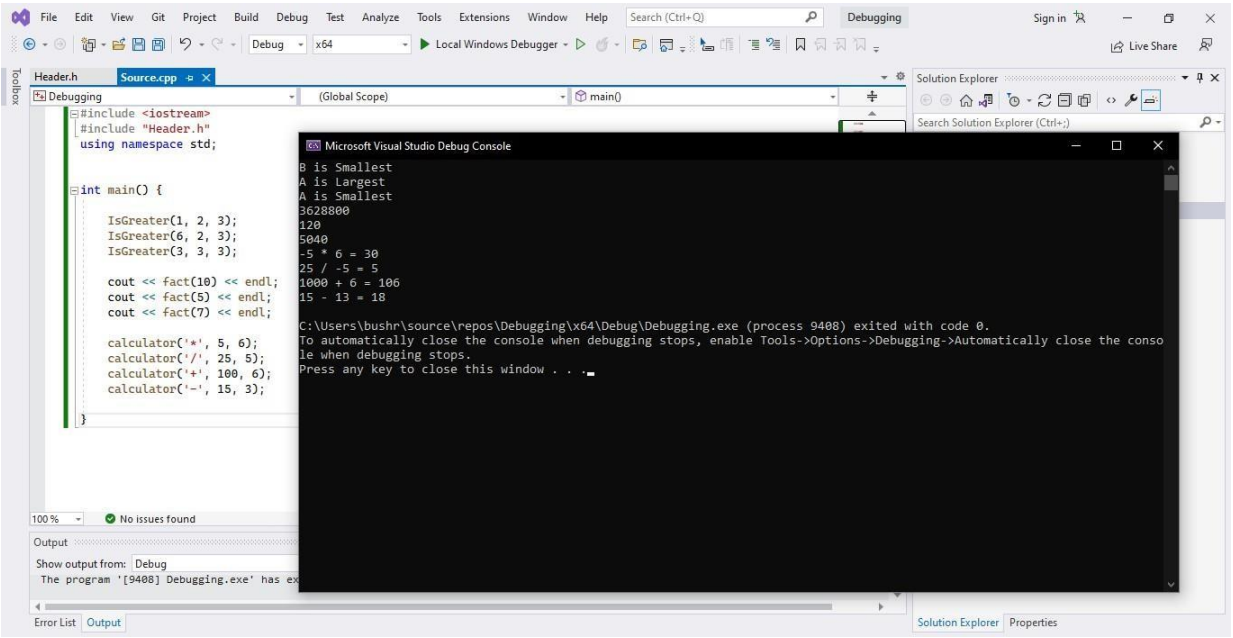


National University of computer and emerging sciences

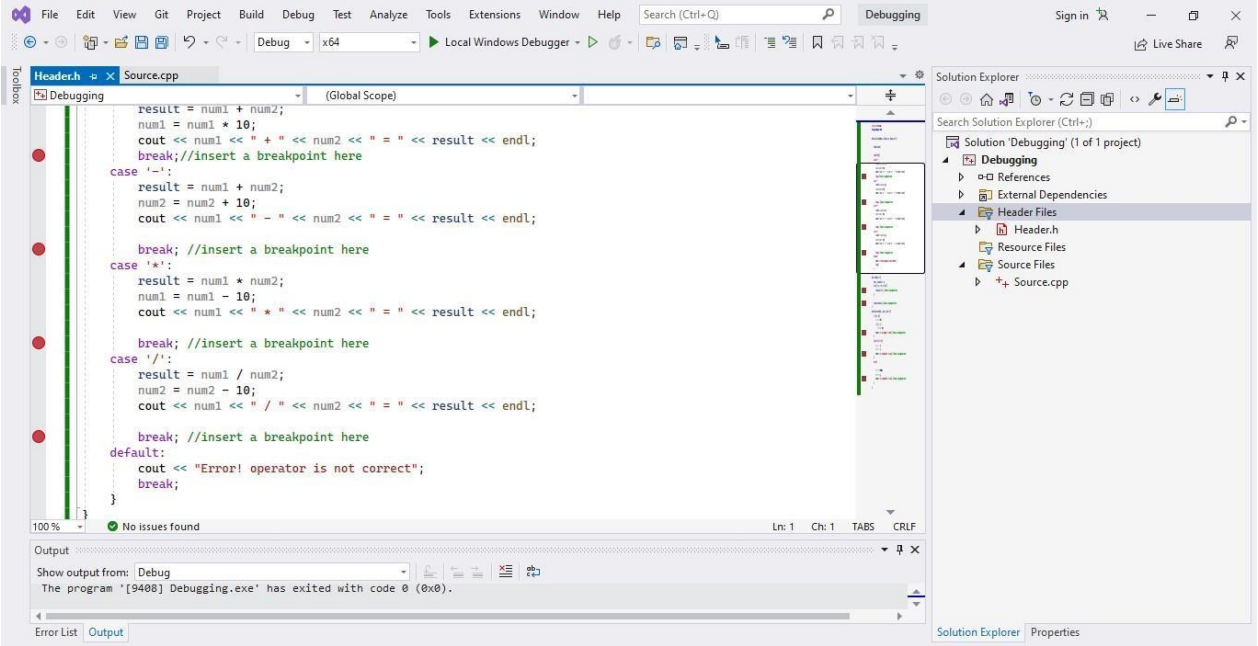
Debugging in Visual Studio

When you debug your app, it usually means that you are running your application with the debugger attached. When you do this, the debugger provides many ways to see what your code is doing while it runs. You can step through your code and look at the values stored in variables, you can set watches on variables to see when values change, you can examine the execution path of your code, see whether a branch of code is running, and so on.

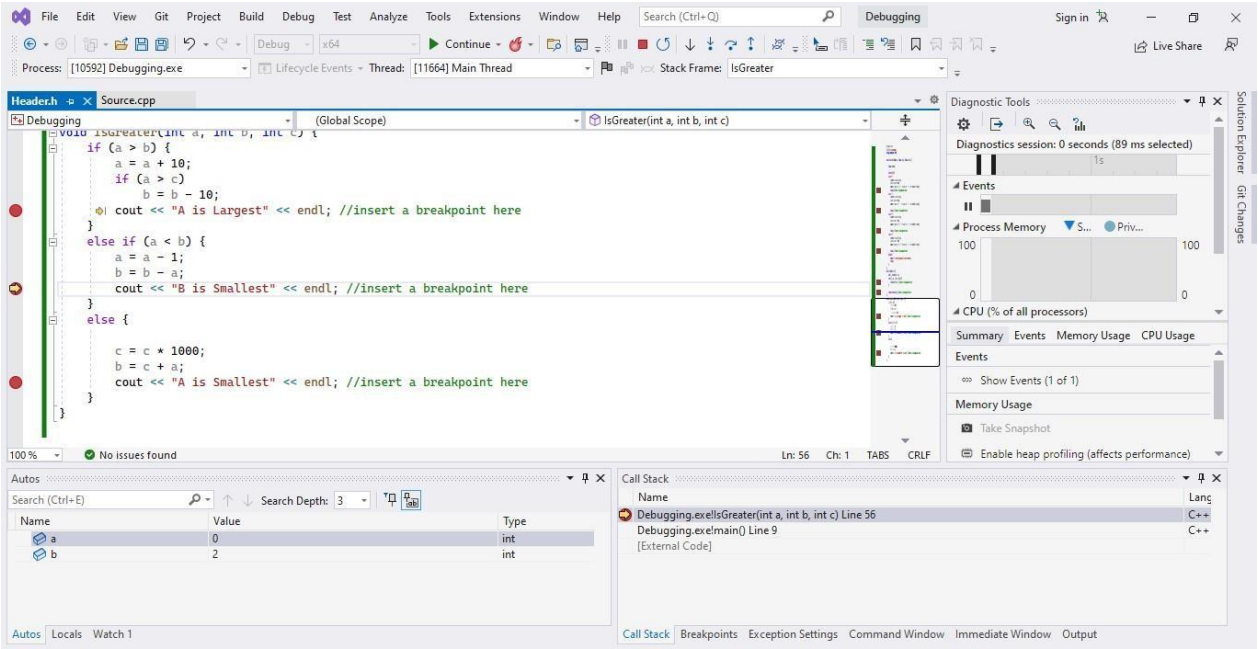
1. Download **debugging_01.cpp** from classroom
2. Now open Visual Studio and create a new C++ project
3. Name your project as **DebugExercise**
4. Create a **source file** and copy the content of the provided files to your created Project files.
5. Run the code, you will see the following output



6. Now add breakpoints in the code at mentioned lines.



7. After adding breakpoints, start debugging the project by clicking on **Debug Icon in the Quick Access toolbar.**



8. Use step in and step out to check the value of each declared variable inside all functions.

National University of computer and emerging sciences

Lab Task

G-Testing Task.

1. Write a program that has an integer array having n elements. The program should have a function that can receive the array and then return the sum of all the elements of the array.
   Function Prototype: int sumArray(int arr[ ],int size)

2. Write a function to find the square root of a number.
   Function Prototype: double squareRoot(const double a)

Note: Run the provided test case file **"test_cases_01.cpp".**

Debugging Task

3. Open "**debugging.cpp**" file and for each function, at all given breakpoints, you need to report the values of each local variable in comments.

Submission Instructions

- Create a new folder with the name **ROLLNO_SEC_LAB01 e.g. i22XXXX_A_LAB01**
- Move your header.h and debugging.cpp file to this newly created directory and compress it into a .zip file.
- Now you have to submit this zipped file on Google Classroom.
- If you don't follow the above-mentioned submission instructions, you will be marked zero.
- Plagiarism in the Lab Task will result in zero marks in the whole category.


National University
of computer and emerging sciences