CMP9133 Advanced Programming
Week 1 – Workshop 0 (Introduction to C++ Development Environment)

Your workshops start in the second week, but you are welcome to use the machines in the lab to try out some basic C++ codes to familiarise yourself with your chosen development environment (Pre-workshop Tasks).

The task instructions are based on Microsoft Visual Studio. After completing the tasks in Microsoft Visual Studio, we recommend utilising Visual Studio Code to do the same. If you have completed the pre-tasks (reading and viewing the videos) or have prior experience with your chosen IDE, we estimate these tasks will take about 45 minutes to complete.

Outcome

- To become familiar with the C++ environment you will used for you workshops.
- To learn, recognize and correct the three types of computer errors: syntax errors run time errors logic errors.
- To learn some basics of an editor and compiler and be able to compile and run existing programs.

Pre-Workshop Watching and Reading:

- Welcome to the Visual Studio IDE https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022
- Create a console calculator in C++ https://docs.microsoft.com/en-gb/cpp/get-started/tutorial-console-cpp?view=msvc-170
- Visual Studio Code - Introductory Videos for C++
- https://code.visualstudio.com/docs/cpp/introvideos-cpp

Procedure

- Ensure you have done the pre-workshop reading above, which will help you understand Visual C++ Environment. If you are new to programming, then the second reading may be difficult to understand but you could still have a look. We don’t expect any one to be new to programming. 😊
- Complete tasks 1.1 through 1.3 in sequence.
    - Tasks 1.1 takes approximately 10 minutes to complete.
    - Tasks 1.2 takes approximately 10 minutes to complete.
    - Tasks 1.3 takes approximately 10 minutes to complete.

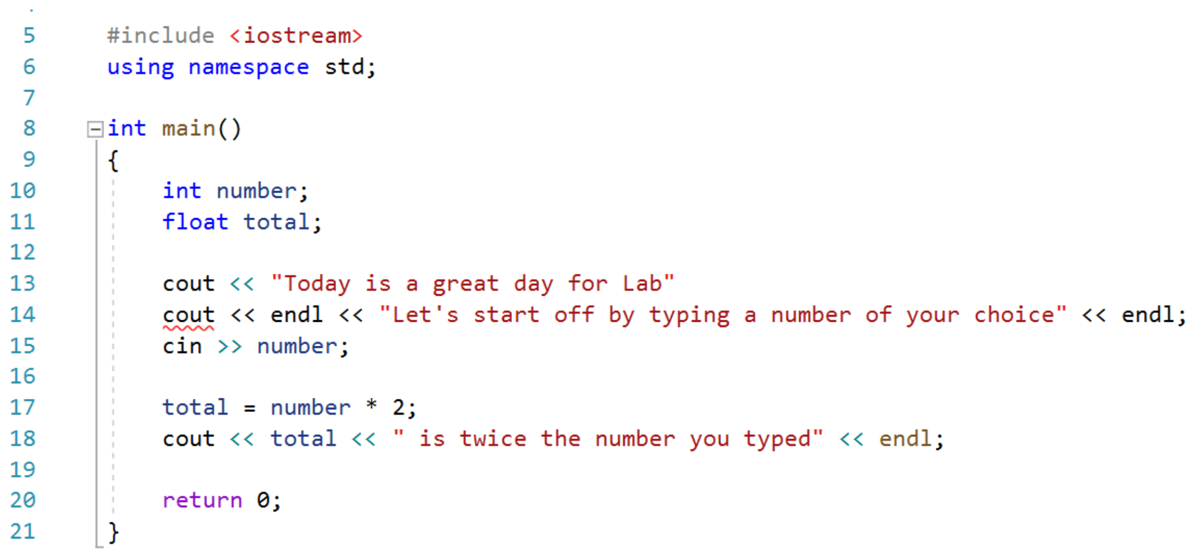
Workshop Tasks

Task 1.1 Compiling a Program with a Syntax Error

- *Exercise 1:* Bring in program semiprob.cpp from the Lab 1 folder.
- *Exercise 2:* Compile the program. Here we have our first example of the many syntax errors that you no doubt will encounter in this course. The error message you receive may be different depending on the system you are using, but the compiler insists that a semicolon is missing somewhere. Unfortunately, where the message indicates that the problem exists, and where the problem occurs may be two different places. To correct the problem, place a semicolon after the line cout << "Today is a great day for Lab".

  Most syntax errors are not as easy to spot and correct as this one.
- *Exercise 3:* Re-compile the program and when you have no syntax errors, run the program and input 9 when asked. Record the output.
- *Exercise 4:* Try running it with different numbers. Record your output. Do you feel you are getting valid output?

The code of semiprob.cpp is as follows:

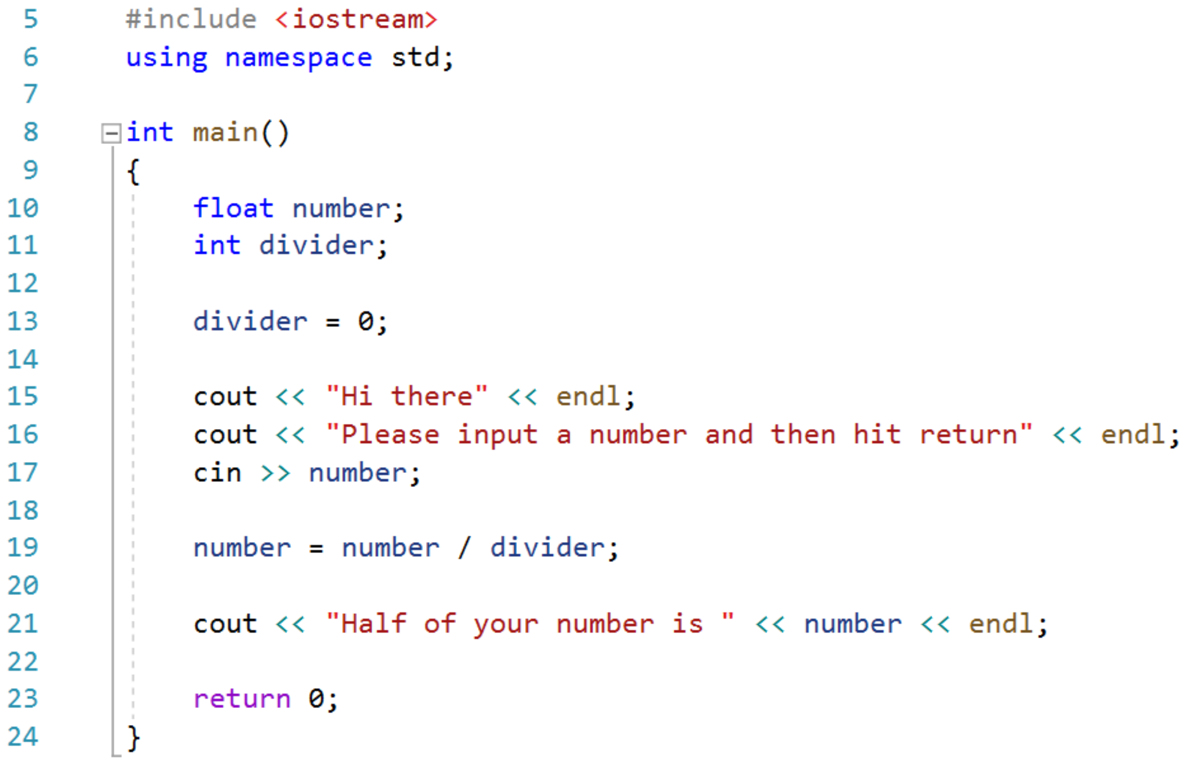


```
#include <iostream>
using namespace std;

int main()
{
    int number;
    float total;

    cout << "Today is a great day for Lab"
    cout << endl << "Let's start off by typing a number of your choice" << endl;
    cin >> number;

    total = number * 2;
    cout << total << " is twice the number you typed" << endl;

    return 0;
}
```

1.2 Running a Program with a Run Time Error

- *Exercise 1:* Bring in program runprob.cpp from the Lab 1 folder.
- *Exercise 2:* Compile the program. You should get no syntax errors.
- *Exercise 3:* Run the program. You should now see the first of several run time errors. There was no syntax or grammatical error in the program; however, just like commanding someone to break a law of nature, the program is asking the computer to break a law of math by dividing by zero. It cannot be done. On some installations, you may see this as output that looks very strange. Correct this program by having the code divide by 2 instead of 0.
- *Exercise 4:* Re-compile and run the program. Type 9 when asked for input. Record what is printed
- *Exercise 5:* Run the program using different values. Record the output. Do you feel that you are getting valid output?

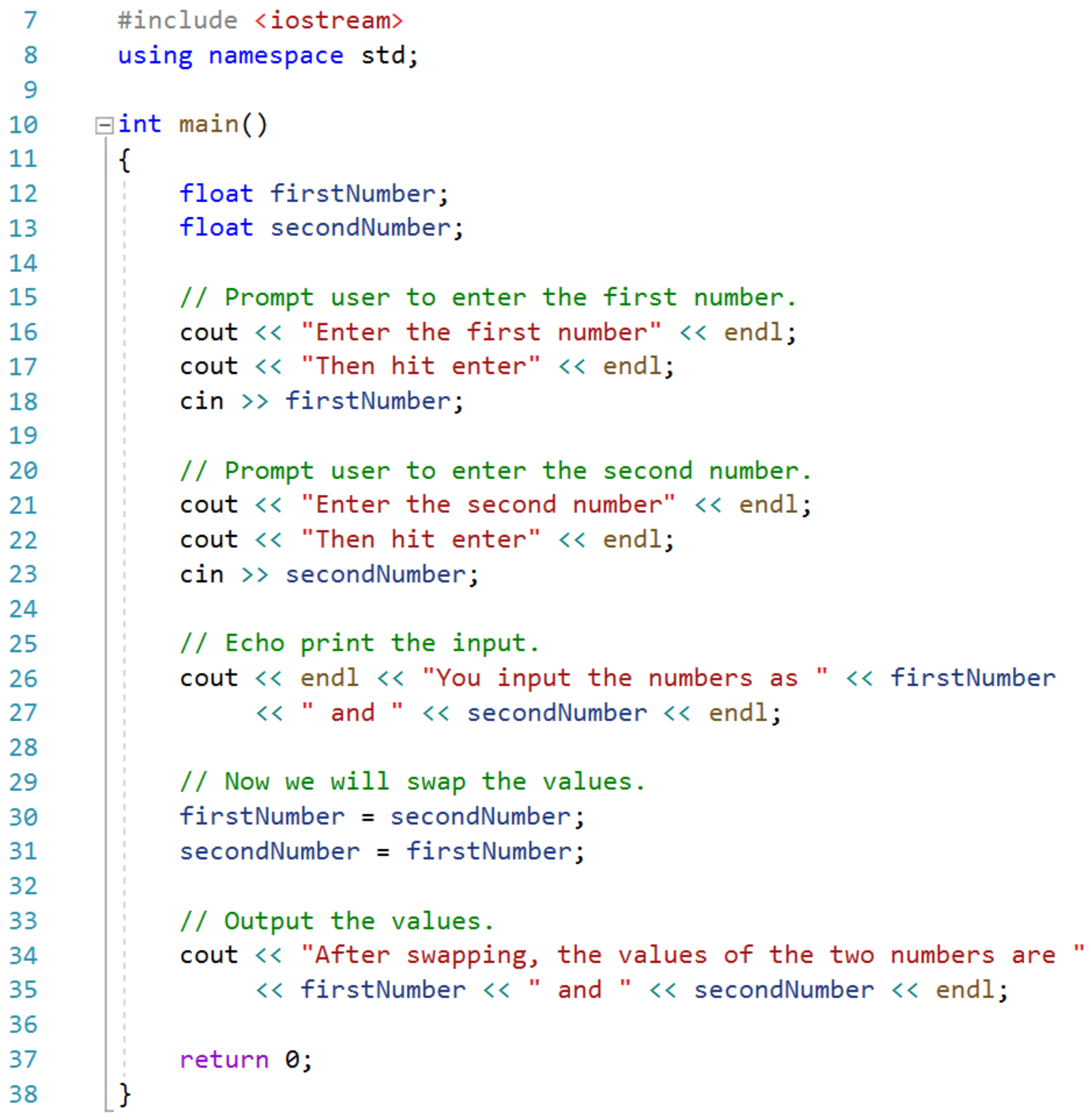
The code of runprob.cpp is as follows.



```cpp
#include <iostream>
using namespace std;

int main()
{
    float number;
    int divider;

    divider = 0;

    cout << "Hi there" << endl;
    cout << "Please input a number and then hit return" << endl;
    cin >> number;

    number = number / divider;

    cout << "Half of your number is " << number << endl;

    return 0;
}
```

1.3 Working with Logic Errors

- *Exercise 1:* Bring in program logicprob.cpp from the Lab 1 folder. The code follows.
- *Exercise 2:* Compile this program. You should get no syntax errors.
- *Exercise 3:* Run the program. What is printed?
- *Exercise 4:* This program has no syntax or run time errors, but it certainly has a logic error. This logic error may not be easy to find. Most logic errors create a challenge for the programmer. Your instructor may ask you not to worry about finding and correcting the problem at this time.



```cpp
#include <iostream>
using namespace std;

int main()
{
    float firstNumber;
    float secondNumber;

    // Prompt user to enter the first number.
    cout << "Enter the first number" << endl;
    cout << "Then hit enter" << endl;
    cin >> firstNumber;

    // Prompt user to enter the second number.
    cout << "Enter the second number" << endl;
    cout << "Then hit enter" << endl;
    cin >> secondNumber;

    // Echo print the input.
    cout << endl << "You input the numbers as " << firstNumber
         << " and " << secondNumber << endl;

    // Now we will swap the values.
    firstNumber = secondNumber;
    secondNumber = firstNumber;

    // Output the values.
    cout << "After swapping, the values of the two numbers are "
         << firstNumber << " and " << secondNumber << endl;

    return 0;
}
```

Remember to repeat these tasks in Visual Studio Code if you have not done so yet.

Appendix

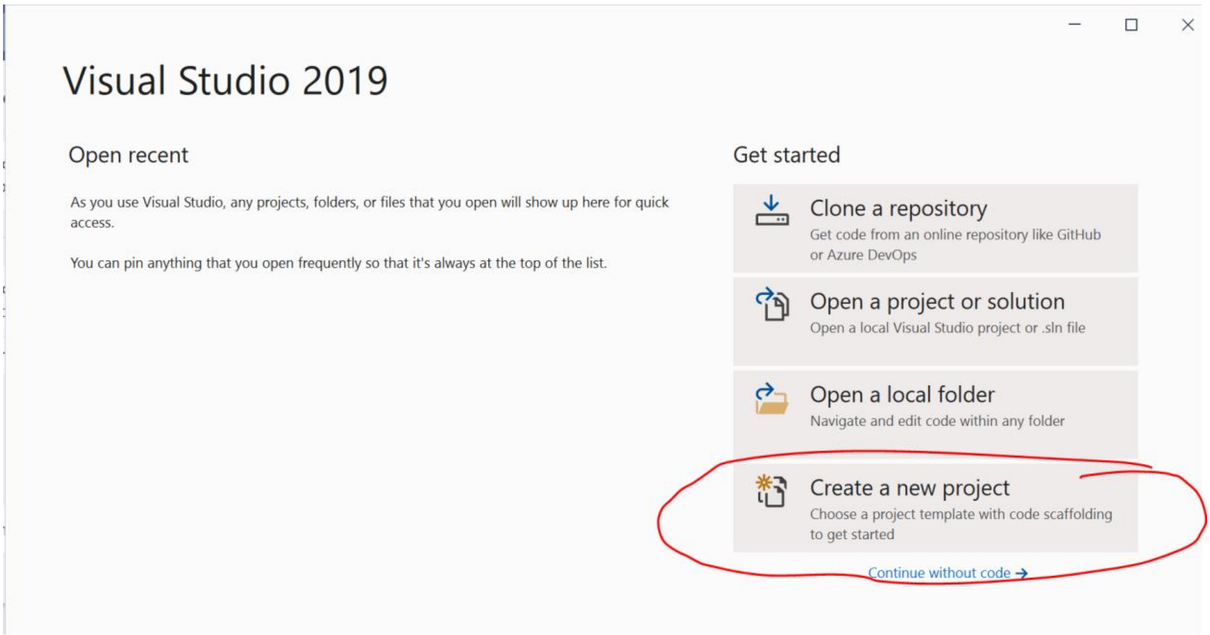
I. Visual C++ Environment

In this first lab, you will become acquainted with a software package that will allow you to create, compile and run C++ programs. Visual C++ is a product of Microsoft and is one component of Visual Studio (a complete set of programming language environments). The following steps walk you through most of the commands needed for this course.

1. Make sure you are logged on the system.

2. Select Start→All Programs→Visual Studio (Your system may have an alternative way of accessing Visual C++, check with your instructor).

Create a new project



Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started

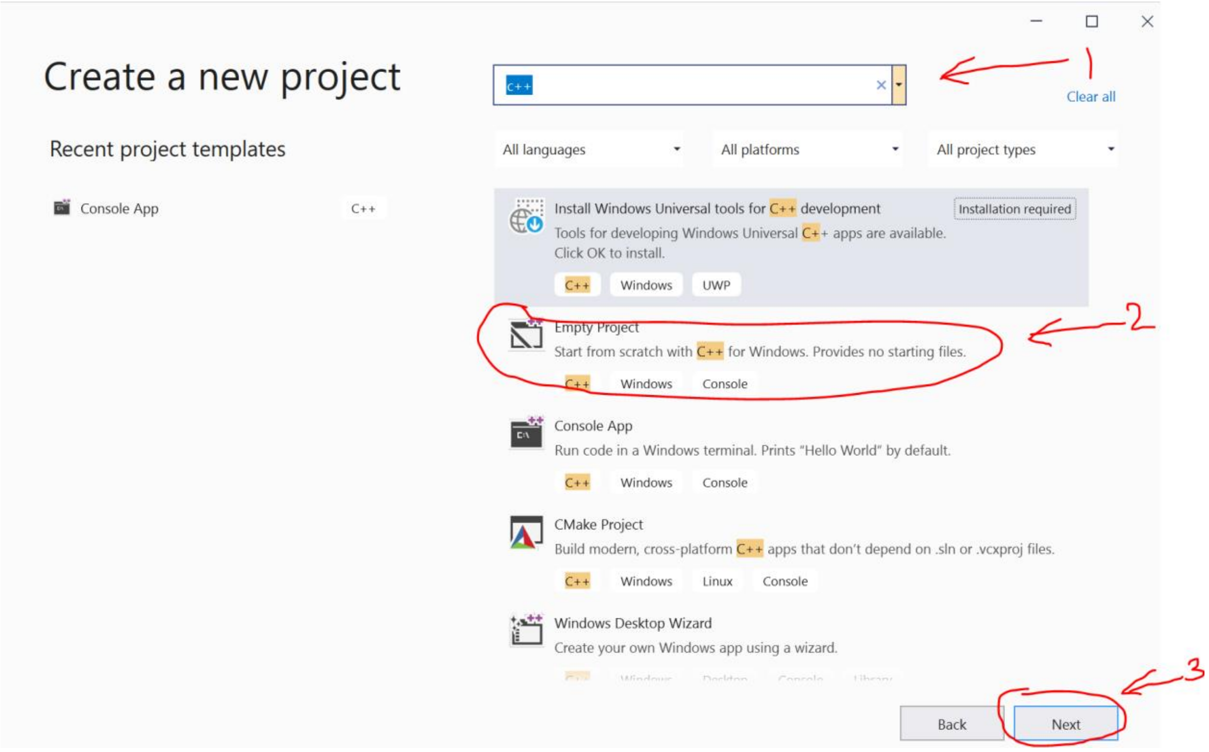
Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Continue without code →



Create a new project

C++ ×

Clear all

Recent project templates

Console App C++

All languages ▾ All platforms ▾ All project types ▾

Install Windows Universal tools for C++ development — Installation required
Tools for developing Windows Universal C++ apps are available. Click OK to install.
C++ Windows UWP

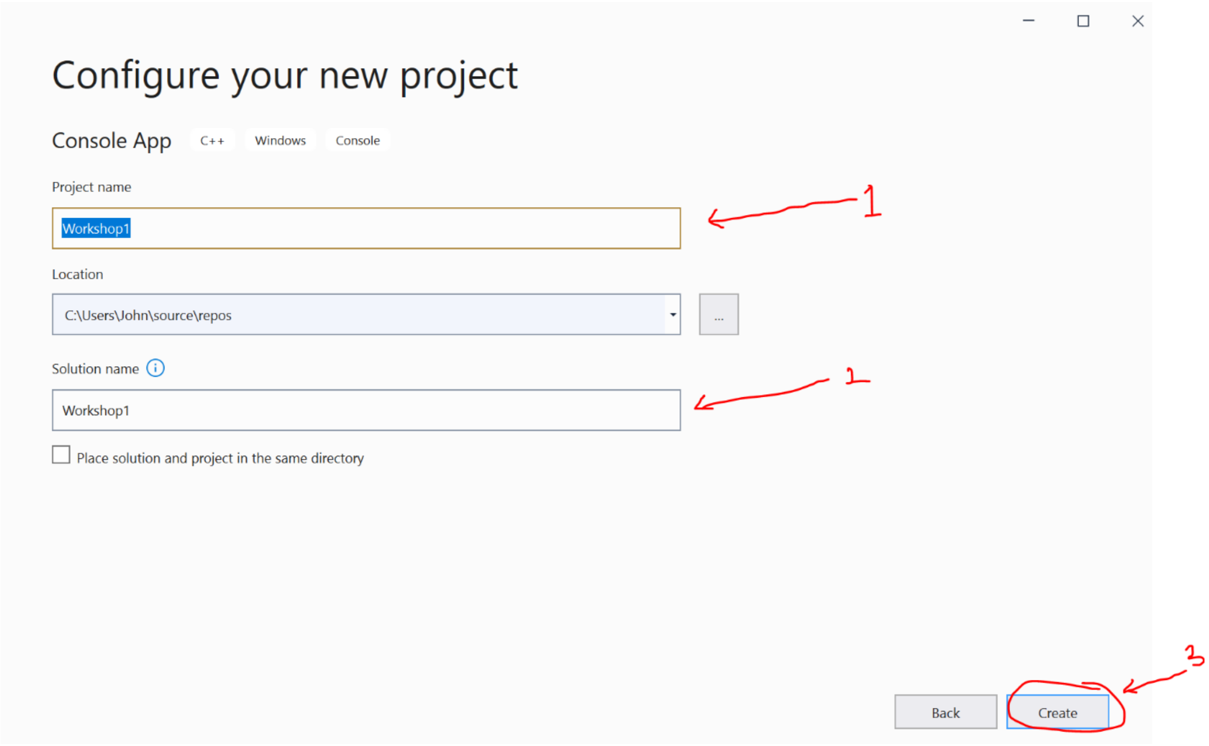
Empty Project
Start from scratch with C++ for Windows. Provides no starting files.
C++ Windows Console

Console App
Run code in a Windows terminal. Prints "Hello World" by default.
C++ Windows Console

CMake Project
Build modern, cross-platform C++ apps that don't depend on .sln or .vcxproj files.
C++ Windows Linux Console

Windows Desktop Wizard
Create your own Windows app using a wizard.
C++ Windows Desktop Console Library

Back Next



Configure your new project

Console App C++ Windows Console

Project name
Workshop1

Location
C:\Users\John\source\repos ▾ ...

Solution name ⓘ
Workshop1

□ Place solution and project in the same directory

Back Create

3. You must import a source file into a project before you will be able to compile and run it.

4. Once you are in the Visual C++ environment you are ready to open or create a project.

II. Importing an Existing File into a Project

1. Start Visual Studio. To start a new project (see previous section for details)
2. Add a file (.cpp) to the created project. In the Solution Explorer window, right-click the Source Files folder. On the menu that pops up select Add, then select Existing Item... You should now see the Add Existing Item dialog box. Browse to the required folder and select the .cpp file you want to add to the project and click the Add button.