

CSCI 1300 CS1: Starting Computing

Fleming, Naidu, Quigley, Fall 2020

Homework 2

Due: Saturday, September 12, by 6 pm

(5 % bonus on the total score if submitted by 11:59 pm September 11)

## Objectives

- Compile and run C++ code
- Take user inputs and produce outputs
- Understand C++ data types
- Perform arithmetic operations

You can find a document called **H2 Background** in the Week 3 Module on Canvas. This document will serve as a reference for some fundamental syntax in C++.

## Submissions

- **C++ files.** All files should be named as specified in each question, and they should compile and run on VSCode to earn full points. TAs will be grading the style of your code and comments. Please see the *Style Guide* file on Canvas and *H2 Background* file on Canvas, both in the Week 3 Module. At the top of each file, write your name with the following format:

```
// CS1300 Fall 2020
// Author: Punith Sandhu
// Recitation: 123 - Favorite TA
// Homework 2 - Problem # ...
```

When you are finished with all the questions, zip all files. Submit the zip file under the assignment **H2\_zip** on Canvas.

- **Code Runner.** The correctness of your program will be graded by Code Runner. You can modify your code and re-submit (press “Check” again) as many times as you need to, until the assignment due date. You can find Code Runner on the page titled **H2-CodeRunner** in the Week 3 Module.

## Questions

### Question 1 (2pt): Hello World

The first program that we usually write in any language we're learning is *Hello, World*. Create a program that prints "Hello, World!" to the screen (the output window in VS Code).

Expected output

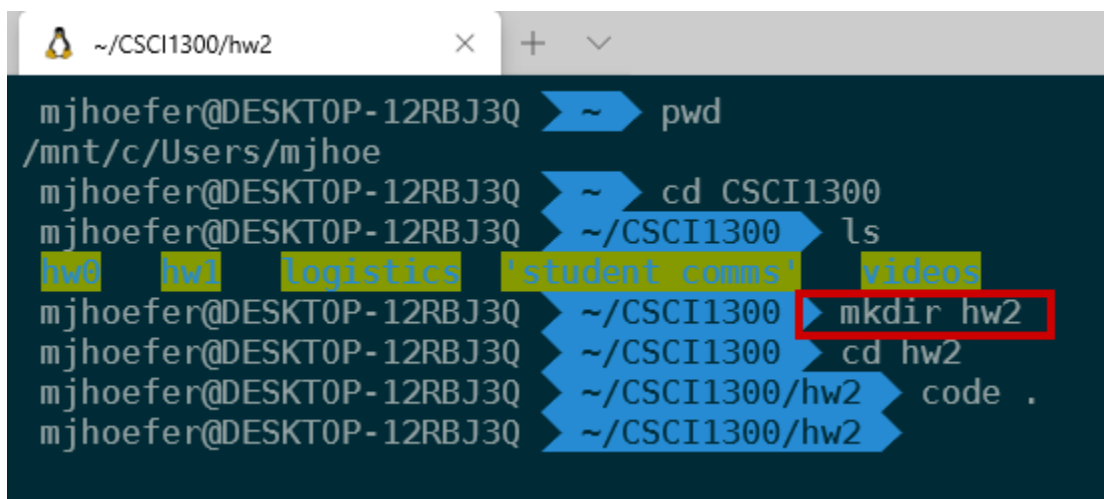
```
Hello, World!
```

The file should be named `helloWorld.cpp`

Here are some suggested steps:

#### Step 1: Create a new folder for hw2

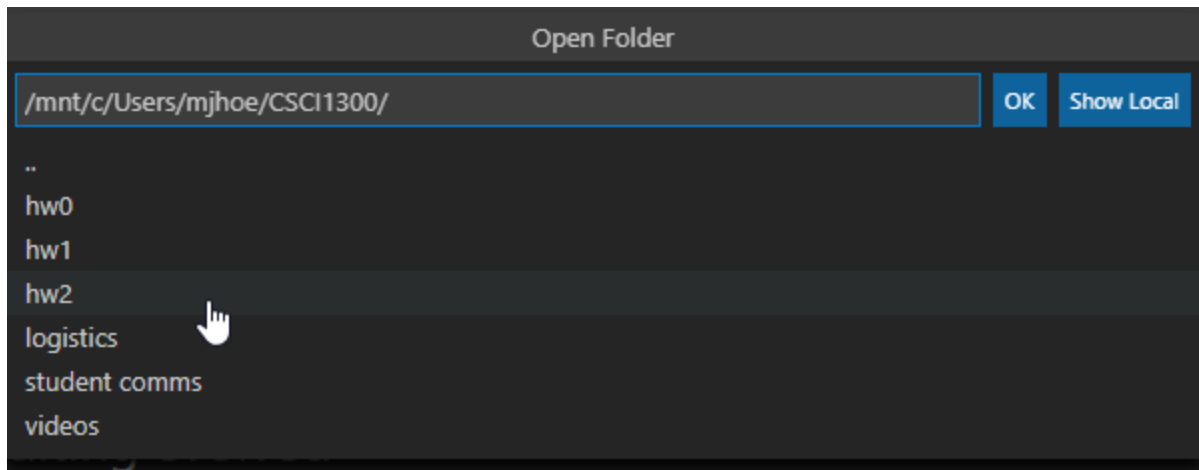
You can create a new folder in the file explorer of your choice or through the terminal. The example below shows the creation of a new folder using the terminal. You can either access this through VS Code (File -> New Terminal) or through the Mac terminal application, or the Windows Ubuntu shell (pictured below).



```
mjhoefer@DESKTOP-12RBJ3Q ~$ pwd
/mnt/c/Users/mjhoe
mjhoefer@DESKTOP-12RBJ3Q ~$ cd CSCI1300
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ ls
hw0  hw1  logistics  'student comms'  videos
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ mkdir hw2
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ cd hw2
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$ code .
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$
```

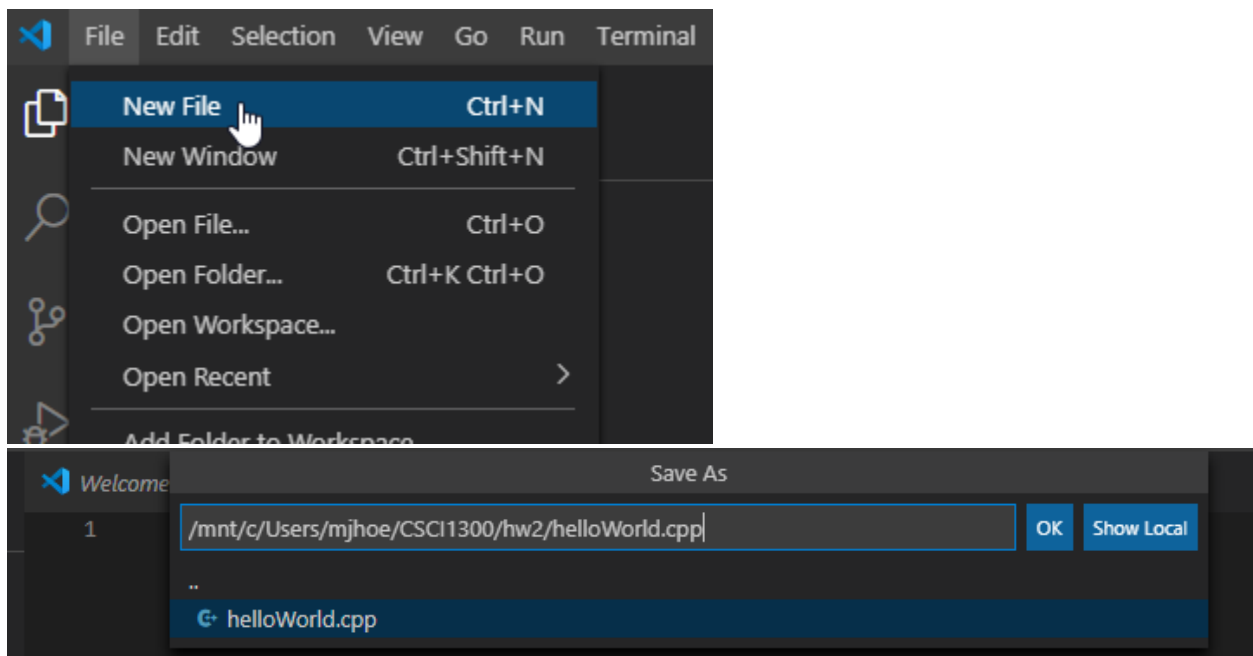
#### Step 2: Open the folder in VS Code

Either open VS Code through the program launcher of your choice (MacOS spotlight, Windows menu, etc) or type "code ." in the terminal when you're in the hw2 directory (as shown in the image above).



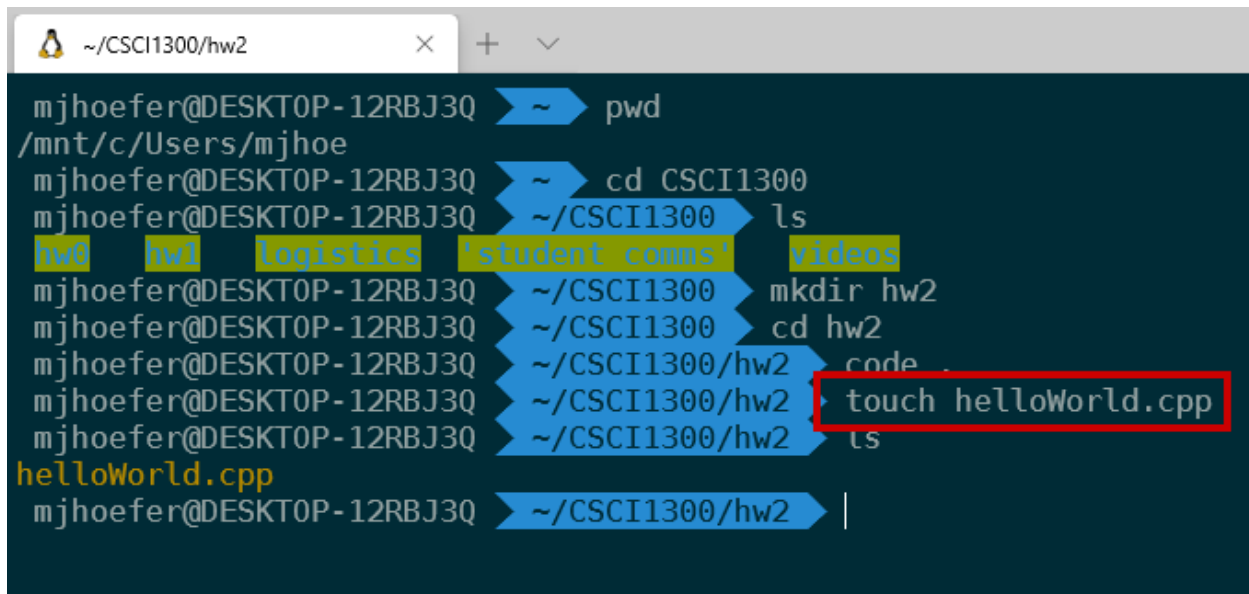
**Step 3: Create a new file in the folder, save it as helloWorld.cpp.**

In VS Code, create a new file. Then save it (File->Save As) as helloWorld.cpp



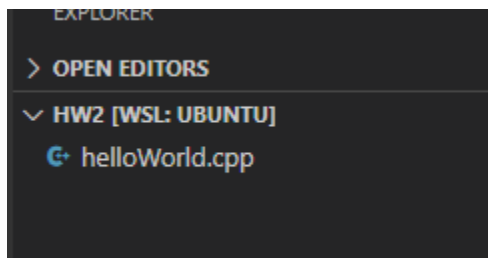
Alternatively, from the terminal, you can simply enter the command:

```
touch helloWorld.cpp
```



```
mjhoefer@DESKTOP-12RBJ3Q ~$ pwd
/mnt/c/Users/mjhoe
mjhoefer@DESKTOP-12RBJ3Q ~$ cd CSCI1300
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ ls
hw0  hw1  logistics  'student comms'  videos
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ mkdir hw2
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300$ cd hw2
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$ code .
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$ touch helloWorld.cpp
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$ ls
helloWorld.cpp
mjhoefer@DESKTOP-12RBJ3Q ~/CSCI1300/hw2$ |
```

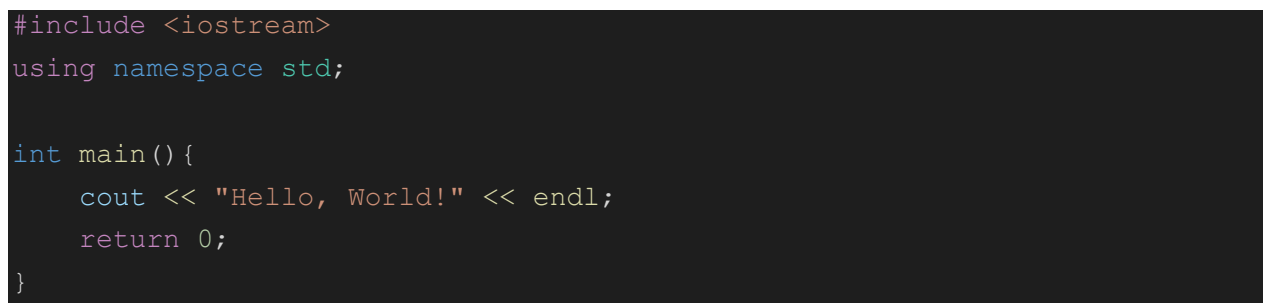
This creates an empty file named `helloWorld.cpp`, which will now show up in the VS Code editor window:



The `.cpp` extension on the filename tells VS Code that the file should be read in the C++ programming language. Once you save it, the lines you write in the file should be color-coded to reflect what they do in the program. This is called syntax highlighting.

#### Step 4: Write Your Code

In VS Code, enter the following hello world program:

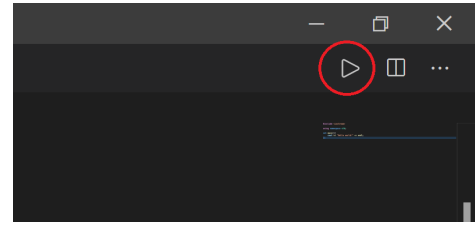


```
#include <iostream>
using namespace std;

int main(){
    cout << "Hello, World!" << endl;
    return 0;
}
```

### Step 5a: Running Your Code

To run the program, click on the icon with the arrow in the top right hand corner. If it works, you should see a new output tab window open at the bottom. You should see [Running] and then a command to run the program and underneath it, the output of our program: "Hello, World!"



**Note:** Many Windows users have reported errors in running their code using the Run button. We believe these errors are the result of some update that has occurred with the Code Runner extension or VSCode within the last year. If this doesn't work for you, check out some solutions that have worked for others on Piazza, or simply run your code using the command line (see below).

### Step 5b: Running Your Code from Command Line

Move to the "terminal" tab (two tabs to the right of "output"). Make sure you are inside the **hmwk2** directory (you can type `pwd` to see where you are). Type:

```
$ g++ helloWorld.cpp -g -std=c++11
```

the `-g` option turns on debugging, which we will use later in the semester, so we should get used to it.

the `-std=c++11` option makes sure that the c++ version used to run the program is c++ 11. If you don't give this option then the default version(which is usually C++98) is used.

This creates an executable called "a.out" (see figure below). You can run it by typing

```
$ ./a.out
```

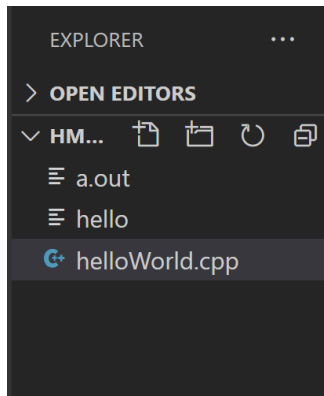
Since no executable name was specified to g++, a.out is chosen by default. You can alternatively use the "-o" option to change the name :

```
$ g++ helloWorld.cpp -g -std=c++11 -o hello
```

creates an executable called "hello" (see figure below). You can run it by typing

```
$ ./hello
```

Notice the output in the same: Hello, World!, followed by the return of the prompt, for new commands.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
1: bash
elizabeth@pareidolia: /mnt/c/Users/Elizabeth/Desktop/CSCI1300/hmwk2$ g++ helloWorld.cpp -g -std=c++11
elizabeth@pareidolia: /mnt/c/Users/Elizabeth/Desktop/CSCI1300/hmwk2$ ./a.out
Hello, World!
elizabeth@pareidolia: /mnt/c/Users/Elizabeth/Desktop/CSCI1300/hmwk2$ g++ helloWorld.cpp -g -std=c++11 -o hello
elizabeth@pareidolia: /mnt/c/Users/Elizabeth/Desktop/CSCI1300/hmwk2$ ./hello
Hello, World!
elizabeth@pareidolia: /mnt/c/Users/Elizabeth/Desktop/CSCI1300/hmwk2$
```

## Step 6: Submit to CodeRunner

Head over to Canvas to the assignment **H2-CodeRunner** in the Week 3 Module. Submit your solution for the first problem and press the Check button. You will see a report on how your solution passed the tests, and the resulting score for the first problem. You can modify your code and re-submit (press “Check” again) as many times as you need to.

Write a program that prints out the following:  
Hello, World!

1. Paste in your code  
from VSCode

Answer. (penalty regime: 0 %)

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     cout << "Hello, World!" << endl;
6 }
7
```

Check



2. Click "Check"

	Expected	Got	
✓	Hello, World!	Hello, World!	✓

Passed all tests! ✓

Correct

Marks for this submission: 2.00/2.00.

If the code runner says "incorrect", click the "show difference" button. It will highlight the differences between the expected output and the output your program produced. These outputs should be exactly the same, including upper/lower case, punctuation, and spaces. Otherwise, it'll mark as incorrect. It's a computer. It needs to be precise.

	Expected	Got	
✗	Hello, World!	hell0, World	✗

Hide differences

Incorrect

Marks for this submission: 0.00/5.00.

### Question 2 (3pt): Hello You!

If a program is more interactive, it's fun! Create a program that takes a name and prints "Hello, <name>!". Your output should be exactly the same as below, including the prompt for the user input.

Expected output (**bold** is user input)

```
Enter your name:  
Malvika  
Hello, Malvika!
```

The file should be named as `helloYou.cpp`. Don't forget to head over to the H2-CodeRunner assignment on Canvas and paste your solution in the answer box!

### Question 3 (5 pt): Temperature conversion

Temperature is commonly described in Celsius, but in the U.S. we tend to use Fahrenheit. Write a program that takes user input for Fahrenheit (as `double`) and converts it into Celsius. The conversion formula is:

$$Celsius = \frac{5}{9} (Fahrenheit - 32)$$

The calculated temperature Celsius values should be formatted with a two-digit precision as shown below. You can use the `setprecision()` function with the `fixed` manipulator from `<iomanip>` library to do so. Fahrenheit value should be printed without any precision formatting applied.

Expected output (**bold** is user input)

```
Enter a temperature in Fahrenheit:  
76  
The temperature 76 degrees Fahrenheit is 24.44 degrees Celsius.
```

The file should be named as `temperature.cpp`. Don't forget to head over to the H2-CodeRunner assignment on Canvas and paste your solution in the answer box!

### Question 4 (10pt): Population

The U.S. Census provides information about the current U.S. population as well as approximate rates of change. Using those rates and the current US population, write a program that takes a current population and computes the projected U.S. population in exactly one year (365 days). If you end up with a non-integer projected population, then round down to the nearest whole person.

Three rates of change are provided:

- There is a birth every 8 seconds
- There is a death every 12 seconds
- There is a new immigrant arriving in the US every 27 seconds



Expected output (**bold** is user input)

```
Enter the current population:
1000000
The population in one year: 3482000
```

The file should be named as `population.cpp`. Don't forget to head over to the H2-CodeRunner assignment on Canvas and paste your solution in the answer box!

### Question 5 (10pt): Convert seconds

Write a program that takes a number of seconds as user input (as an `integer`) and converts it to hours, minutes, and seconds as shown below. You should convert the amount of time in such a way that maximizes the whole numbers of hours and minutes.

Expected output 1 (**bold** is user input)

```
Enter a number of seconds:
60
0 hour(s) 1 minute(s) 0 second(s)
```

Expected output 2 (**bold** is user input)

```
Enter a number of seconds:
3671
1 hour(s) 1 minute(s) 11 second(s)
```

The file should be named as `convertSeconds.cpp`. Don't forget to head over to the H2-CodeRunner assignment on Canvas and paste your solution in the answer box!

### Question 6 (10pt): Calculating the volume and surface area of a sphere

Create a program that takes the radius of a sphere as a floating-point value input (as `double`), and prints both the volume and surface area of the sphere with the given radius. The formulas are:

$$volume = \frac{4}{3}\pi r^3$$
$$surfaceArea = 4\pi r^2$$

For  $\pi$ , use `M_PI` from `#include <cmath>`

Expected output (**bold** is user input)

```
Enter a radius:
3.3
Volume: 150.533
Surface area: 136.848
```

The file should be named as `sphereVolumeArea.cpp`. Don't forget to head over to Code Runner on Canvas and paste your solution in the answer box!

## Homework 2 checklist

Here is a checklist for submitting the assignment:

1. Use your solutions in VS Code to complete the **H2-CodeRunner** assignment on Canvas  
(Modules → Week 3)
2. Submit one zip file to the **H2\_zip** assignment on Canvas. The zip file should be named, **hmwk2\_lastname.zip**. It should have the following 6 files:
  - **helloWorld.cpp**
  - **helloYou.cpp**
  - **temperature.cpp**
  - **population.cpp**
  - **convertSeconds.cpp**
  - **sphereVolumeArea.cpp**

## Homework 2 points summary

Criteria	Pts
CodeRunner (problem 1 - 6)	40
C++ files zip submission (compiles and runs, style and comments)	15
Recitation attendance (Week 3)*	-20
Total	55
5% early submission bonus	+5%

\* if your attendance is not recorded, you will lose points. Make sure your attendance is recorded on Canvas.