

# Using replit.com

Many of the examples in this course are available at [replit.com](https://replit.com). This is a publically available on-line editor-debugger-and-compiler that allows a user to create and edit a repository, or collection, of programs. You can read more about this site on [Wikipedia](https://en.wikipedia.org/wiki/Replit). This site allows users to upload or generate code; edit, debug, compile and execute code; and allows other users to compile and execute that code, or alternatively, download that code, or more usefully, fork that code into their own repository, where the user can now edit, debug, compile and execute code in their repository without changing the code in the original user's repository.

To experience replit.com, first go to the [Hello World!](#) repl from this course. You will be welcomed with a rectangular panel with a large Play button, as shown in Figure 1.

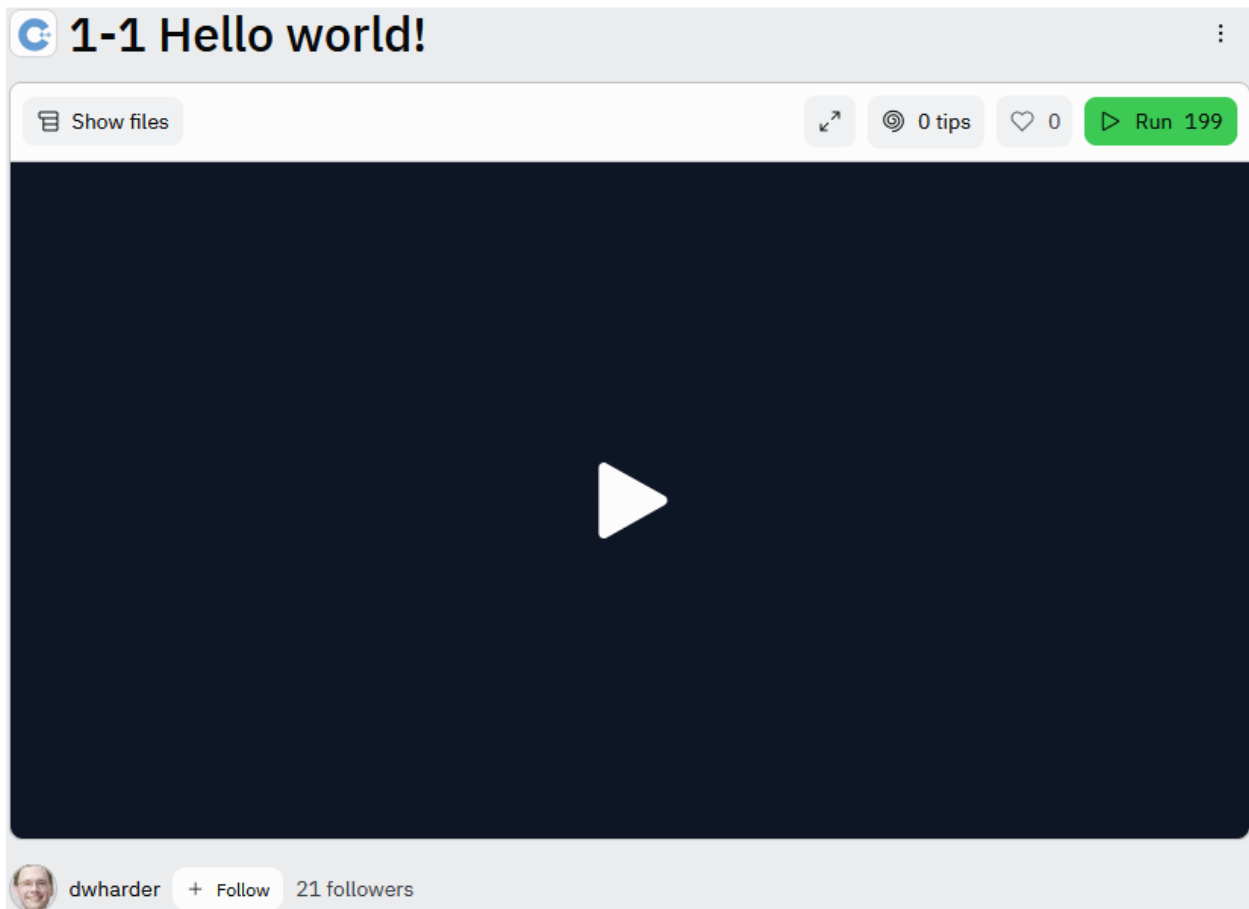


Figure 1. The Hello World! repl.

To compile and execute this code, select the large, friendly play button. When you select this button, you will see the panel replaced by:

```
> clang++-7 -pthread -std=c++11 -o main main.cpp
> ./main
Hello World!
>
```

The first line is the call to the compiler to compile the file `main.cpp` into an executable. The gold symbol is a prompt, meaning that it is prompting you to enter commands for the operating system. The first word, `clang++-7`, is the name of the compiler. Up until now, most of you have only interacted with programs through a graphical user interface (or GUI) where you execute a program by selecting the appropriate icon. Internally, however, each icon is associated with an executable file with a name. By typing the name of that executable at the prompt, the operating system will launch that program.

## Compiling and executing a program

In this case, the compiler program takes options and input:

1. The first option is `-pthread`. This is beyond the scope of this course, but this tells the compiler to use the pthread library.
2. The second option is `-std=c++11`. This tells the compiler to use the C++ 2011 standards. There are many different versions of C++, and every few years, a new version is created, which has new features. There are older and more recent standards (`c++98`, `c++03`, `c++11`, `c++14`, `c++17`, `c++20` and `c++23`). We use C++ 2011 because it has the features we require. Changes beyond C++ 2011 do not affect this course.
3. The third option is `-o main`. This tells the compiler that the name of the executable should be `main`. If you do not give the compiler an output or executable name, the default will be either `a.exe` or `a.out`, depending on whether you are using Windows or Linux/macOS, respectively.
4. The final input is the name of the file to be compiled. In this case, `main.cpp`. This is a text file containing the program.

The replit environment automatically executes this command. If there were warnings or error messages, these would be printed immediately below the command; however, in this case, the compilation is successful, so at the next prompt appears the name of the executable `./main`. The prefixed `./` is specific to the Linux/macOS environment, but is required to execute that executable. That command is also automatically executed, and you can see the output: the text `Hello world!`

At this point, you see a third prompt with no text. You could, if you wished, type another command. For now, the most you might consider doing is explicitly typing `./main` and then pressing Enter. This will execute the executable a second time.

## Viewing the source code

If you select the  Show files button in the top left-hand corner of the panel, this will open a left-hand menu that lists all of the files in this project, as is shown in Figure 2.

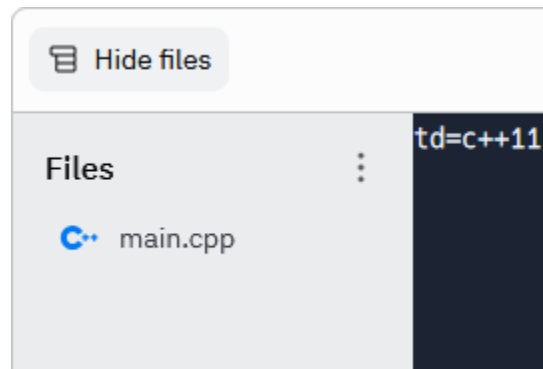


Figure 2. The files in the Hello World! repl project.

In this case, there is only one text file: `main.cpp`, which is shown in Figure 3.

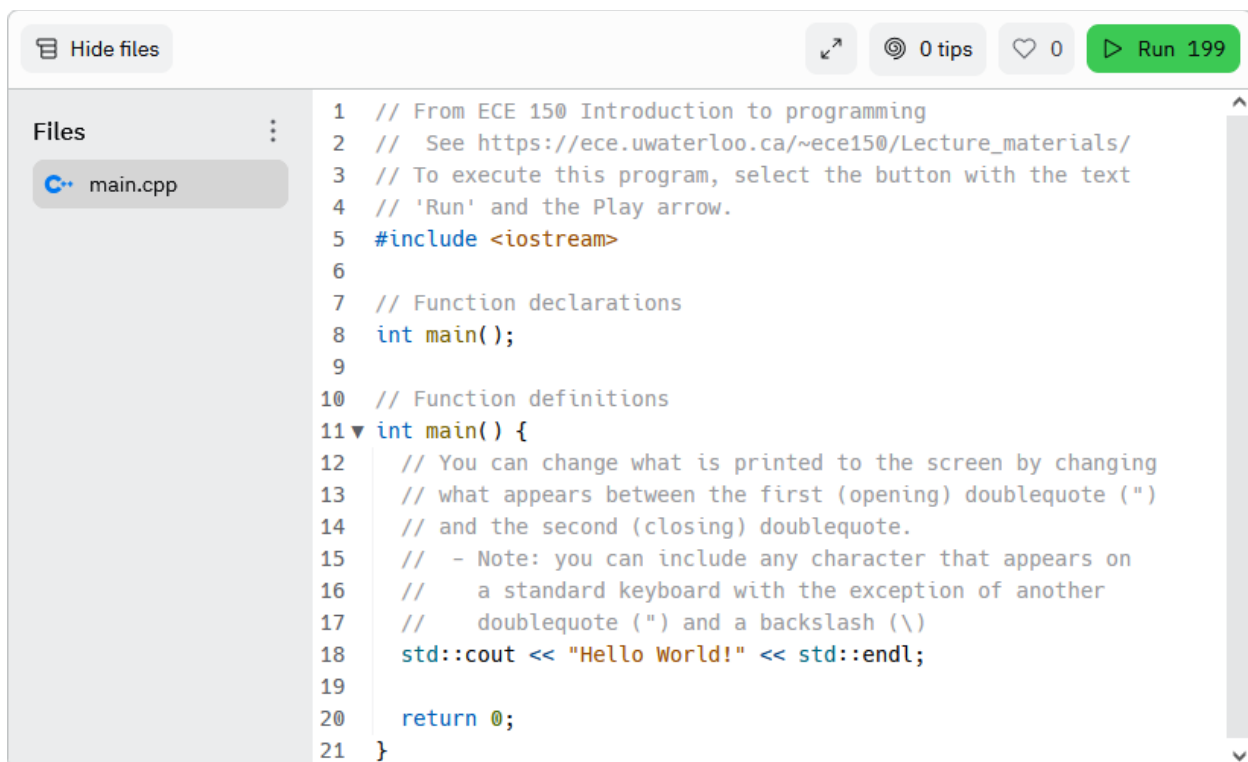


Figure 3. The `main.cpp` source file.

Each line is given a line number, and this line number doesn't actually appear in the text file. It will be used so that when the compiler tells you there is an error on line, for example, 14, you can quickly find that line in this file.

All text that appears after two slashes, `//`, are *comments*. This is English text that is used to help explain the program to the reader. Any comments do not affect compilation. You will notice that


this editor uses color highlighting: all comments appear in gray. Other aspects of the program are colored either black, blue, brown, gold or teal. At this point, you can only view the source code; you cannot edit it. To edit source code, you must create a repl account.

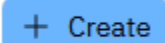
## Creating an account and getting a replit.com repository

First, go to [replit.com](https://replit.com) and in the top right-hand corner, select the Sign Up button. Alternatively, go directly to the [sign-up page](#).


## Logging on

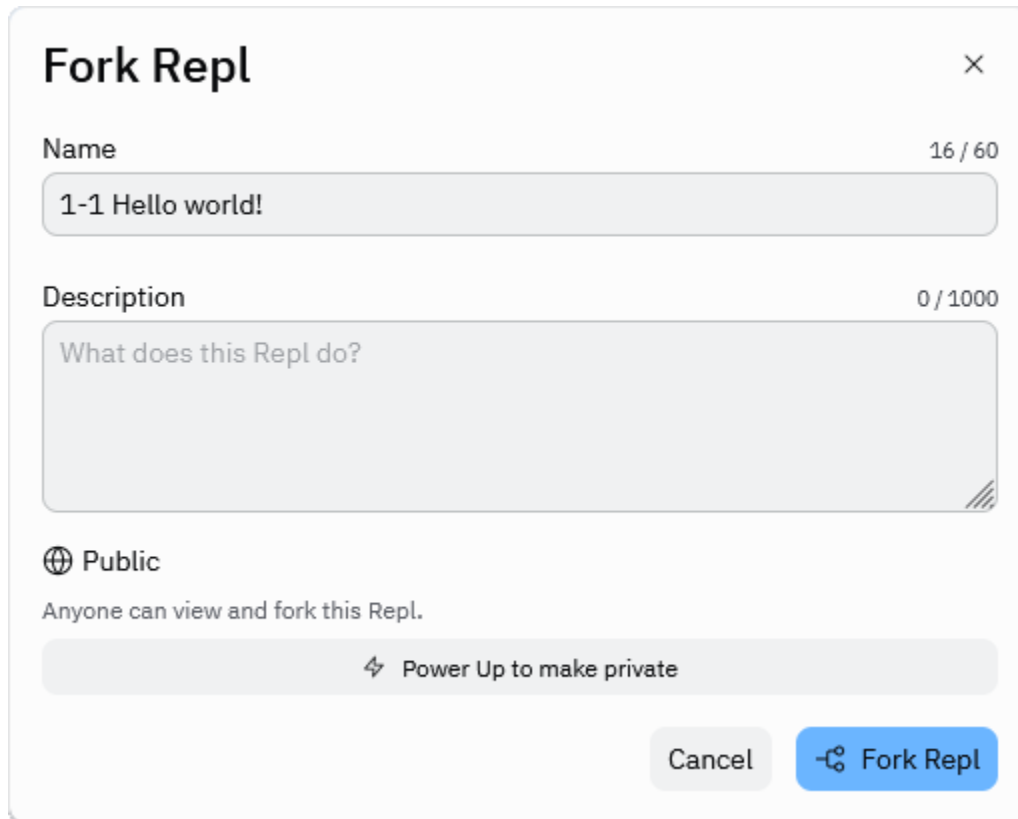
When you log on, you will see a welcome page that has the left-hand menu open by default,

although you can collapse this by selecting the menu button . In the main panel, you will see My Repls which, at this point, should be empty. You can create a project by selecting the

 button either below this section or from the left-hand menu. Creating repl projects is discussed below. Alternatively, once you have logged on, you can visit any other publicly visible repl, you can then compile and execute that code, or alternatively, you can fork that code into your repository. This, too, is discussed below.

## Forking repls

Before we create a personal repl, we will discuss [forking](#) existing repls. For example, once you are logged on, you can return to the Hello World! program and in the top right-hand corner, select the Fork button . This will bring up the dialog shown in Figure 4. You will note that this fork is public, meaning anyone can see this repl and the changes you make to it. The dialog encourages you to select Power Up if you wish to make this forked repl private, which we discuss in the next section.



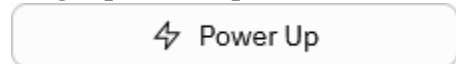
The image shows a 'Fork Repl' dialog box with a close button (X) in the top right corner. It contains a 'Name' field with the text '1-1 Hello world!' and a character count of '16 / 60'. Below it is a 'Description' field with the placeholder text 'What does this Repl do?' and a character count of '0 / 1000'. Under the description, there is a globe icon followed by the word 'Public' and the text 'Anyone can view and fork this Repl.'. At the bottom of the dialog, there is a button with a lightning bolt icon and the text 'Power Up to make private'. In the bottom right corner, there are two buttons: a grey 'Cancel' button and a blue 'Fork Repl' button with a fork icon.

Figure 4. Forking a repl.

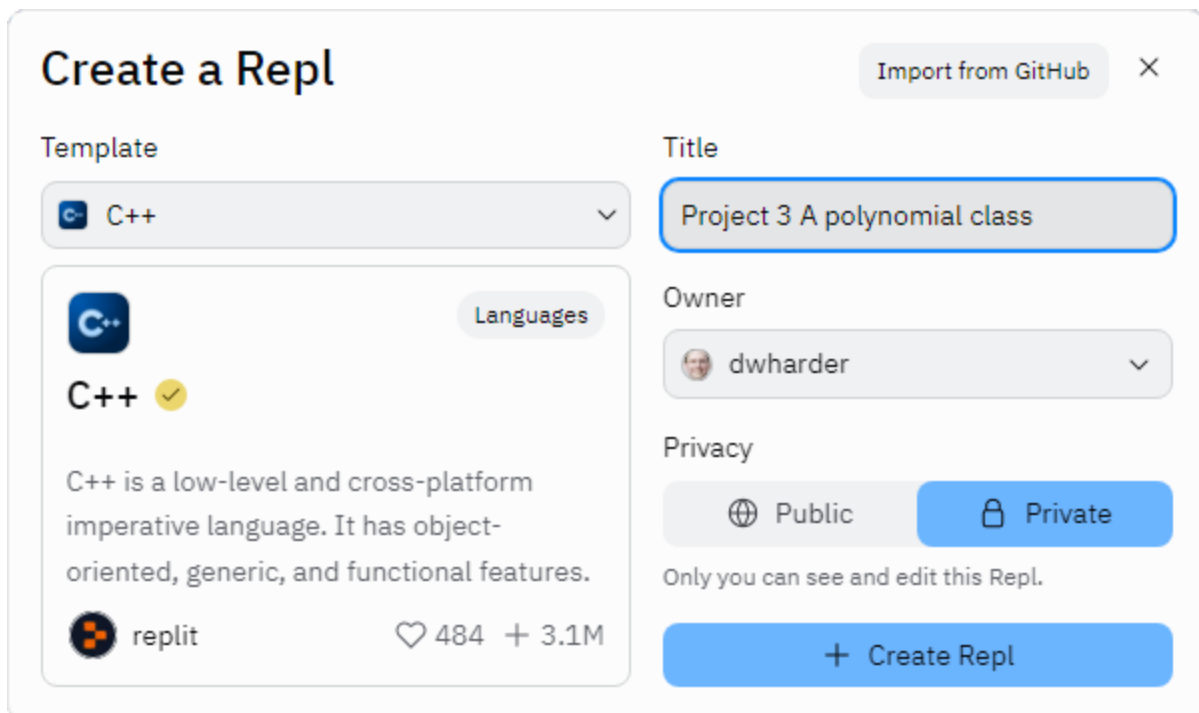
## Private repls

With a free account, you can create repls, but these are by default public, meaning anyone else in the world can search for, find and fork them. If you do a project in ECE 150 and you happen to do it at replit.com, and someone else finds it and copies it, this is your responsibility: you are as much at fault as the person who copied your code. Remember, if you leave a laptop in your unlocked car, or even in a locked car if it is clearly visible, and it is stolen, or you use a poor password, or use the same password at many different sites, both professionally and privately, then if your accounts are broken into, this is your fault. You will be charged with a violation of Policy 71, as you made it possible for another person to easily access your code even if you did not explicitly send your code to that other individual.

To get private repls, from the left-hand menu on the launch page, select the Power Up button



and then select an appropriate plan. This author uses the 75 USD/year Hacker Plan; however, they do offer a monthly payment schedule, too. Now, when you create a repl, you can identify it as being Private, as is shown in Figure 5. When you fork a repl, there will be an equivalent button to select. When you look at your repls, you will note that the lock symbol will also appear next to the repl listing.



The screenshot shows the "Create a Repl" interface. At the top right is an "Import from GitHub" button with a close icon. The form is divided into two main sections: "Template" and "Title".

- Template:** A dropdown menu is set to "C++". Below it is a card for the C++ template, featuring the C++ logo, a "Languages" tag, the text "C++" with a checkmark, a description: "C++ is a low-level and cross-platform imperative language. It has object-oriented, generic, and functional features.", the Replit logo, and statistics: "484 + 3.1M".
- Title:** A text input field contains "Project 3 A polynomial class".
- Owner:** A dropdown menu shows the user "dwharder".
- Privacy:** Two buttons are visible: "Public" (with a globe icon) and "Private" (with a lock icon). The "Private" button is highlighted in blue. Below these buttons, the text reads "Only you can see and edit this Repl."
- Bottom:** A large blue button with a plus sign and the text "+ Create Repl".

Figure 5. Creating a private repl.

## Comments on replit.com and integrated development environments

If you have never programmed before, perhaps it is best to start by using `replit.com` or some other similar on-line editor-debugger-and-compiler environment. This allows you to quickly start authoring, testing, debugging and completing assignments and projects, be they personal or for an evaluation. If you are in electrical engineering or in some other field of engineering other than computer or software engineering or some other field altogether outside of engineering, you may prefer to complete this course using only `replit.com`; however, if you ever wish to enter into software development, you will need to understand more general desktop [integrated development environments](#) (IDES). This is a decision you must make yourself, but [Virtual Studio Code](#) is a good place to start for Windows, [Xcode](#) is a good for users of macOS, and [CLion](#) is popular cross platform. Another popular cross-platform ide is [Eclipse](#); however, Eclipse is designed to be as general as possible, and thus is more difficult specifically. If you pick such an IDE, you must make sure that the compiler you are using is using the C++ 2011 standard.