Contents

[Compile-time Errors 1](#_Toc56970521)

[Link-time Errors 2](#_Toc56970522)

[Run-time Errors 2](#_Toc56970523)

[Logic Errors 3](#_Toc56970524)

[Summary 3](#_Toc56970525)

## Compile-time Errors

When we are writing C++ programs, the compiler is our first line of defense against errors.

There are two types of compile-time errors:

* **Syntax errors**: Errors that occur when we violate the rules of C++ syntax.
* **Type errors**: Errors that occur when there are mismatch between the types we declared.

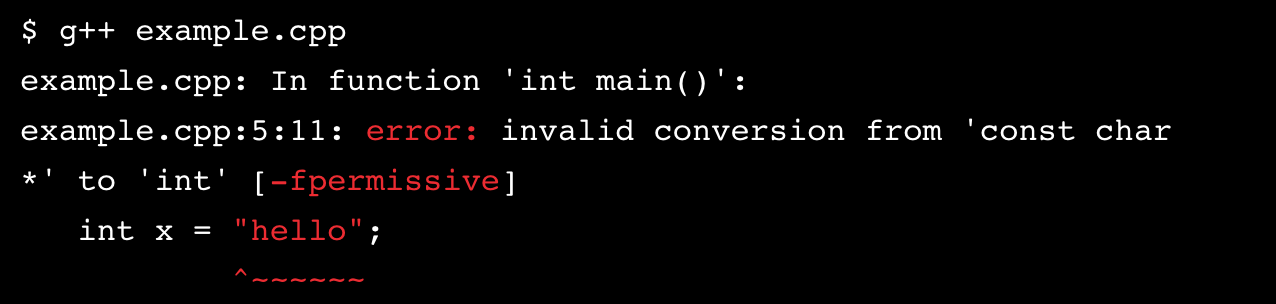
Some common syntax errors are:

* Missing semicolon;
* Missing closing parenthesis ), square bracket ], or curly brace }

Some common type errors are:

* Forgetting to declare a variable
* Storing a value into the wrong type

Here’s an example of a compile-time error message:



The compiler will tell us where (line number) it got into trouble and its best guess as to what is wrong.

## Link-time Errors

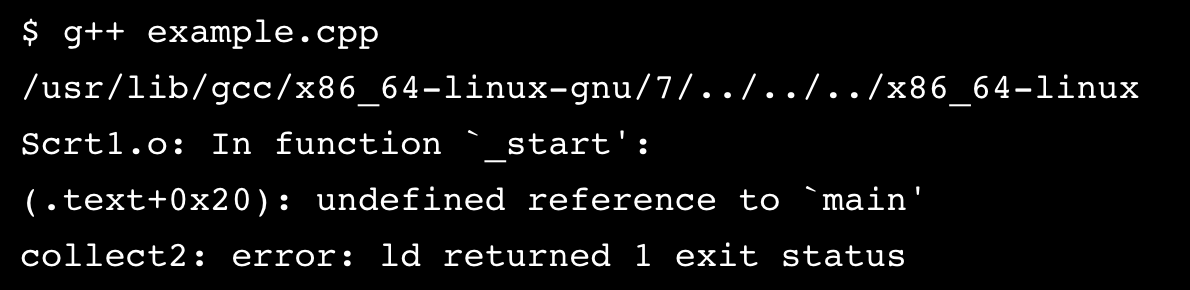
Sometimes the code compiles fine, but there is still a message because the program needs some function or library that it can’t find. This is known as a link-time error.

As our program gets bigger, it is good practice to divide the program into separate files. After compiling them, the linker takes those separate object files and combines them into a single executable file. Link-time errors are found by the linker when it is trying to combine object files into an executable file.

Some common link-time errors:

* Using a function that was never defined (more on this later)
* Writing Main() instead of main()

Here’s an example of a link-time error message:



These errors are more hard to find, but remember, [Google](https://www.google.com/) is your friend! Here, a good Google search would be: “C%20%20 undefined reference to main”.

## Run-time Errors

If our program has no compile-time errors and no link-time errors, it’ll run. This is where the fun really starts.

Errors which happen during program execution (run-time) after successful compilation are called run-time errors. Run-time errors occur when a program with no compile-time errors and link-time errors asks the computer to do something that the computer is unable to reliably do.

It happens after we give the ./ execute command:

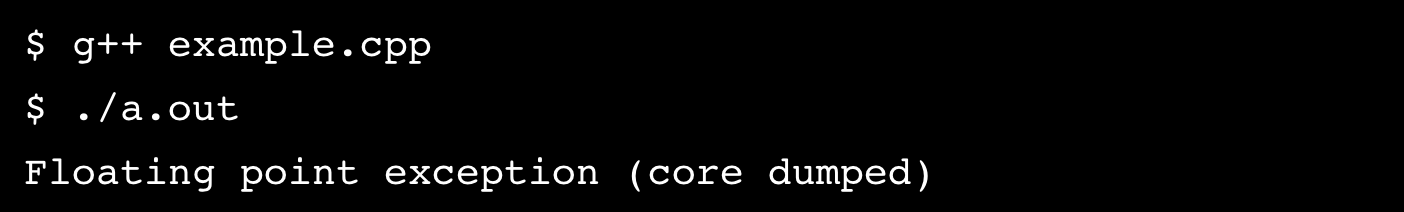
./a.out

Some common run-time errors:

* Division by zero also known as *division error*. These types of error are hard to find as the compiler doesn’t point to the line at which the error occurs.
* Trying to open a file that doesn’t exist

There is no way for the compiler to know about these kinds of errors when the program is compiled.

Here’s an example of a run-time error message:



## Logic Errors

Once we have removed the compile-time errors, link-time errors, and run-time errors, the program runs successfully. But sometimes, the program doesn’t do what we want it to do or no output is produced. Hmmm…

These types of errors which provide incorrect output, but appears to be error-free, are called logical errors. These are one of the most common errors that happen to beginners and also usually the most difficult to find and eliminate.

Logical errors solely depend on the logical thinking of the programmer. Your job now is to figure out why the program didn’t do what you wanted it to do.

Some common logic errors:

* Program logic is flawed
* Some “silly” mistake in an if statement or a for/while loop

**Note:** Logic errors don’t have error messages. Sometimes, programmers use a process called [test-driven development (TDD)](https://en.wikipedia.org/wiki/Test-driven_development), a way to give logic errors error messages and save yourself a lot of headaches!

## Summary

Finding bugs is a huge part of a programmer’s life. Don’t be intimidated by them… embrace them. Errors in your code mean you’re trying to do something cool!

In this lesson, we have learned about the four types of C++ errors:

* **Compile-time errors:** Errors found by the compiler.
* **Link-time errors:** Errors found by the linker when it is trying to combine object files into an executable program.
* **Run-time errors:** Errors found by checks in a running program.
* **Logic errors:** Errors found by the programmer looking for the causes of erroneous results.

Remember, [Google](https://www.google.com/) and [Stack Overflow](https://www.stackoverflow.com/) are a programmer’s best friends. For some more motivation, check out this blog post: [Thinking About Errors in Your Code Differently](https://news.codecademy.com/errors-in-code-think-differently).

We wish you the best of luck in your bug-squashing journey.