

Strongly vs Weakly Typed Languages

C++ is a *strongly typed* language. That means that any time an operation is performed on operands which would result in a nonsensical value, the compiler will give you an error! The errors that the compiler generates if you attempt to perform an operation on values with incompatible types is called a *type error*. For example,

```
int one{"5" / 5};  
std::cout << "one: " << one << "\n";
```

will not compile and you will get a type error. How could that code possibly compile? What does it even mean to divide a string by a number?

Strongly typed languages are more reliable than their counterparts, weakly typed languages. (*Note:* This statement is not meant to imply that weakly typed languages are in every way inferior to strongly typed languages). Weakly typed languages, like Python and JavaScript, have complex rules defining their behavior for performing operations on otherwise incompatible types. For instance, in Python, if you attempt to divide a string by a second number (like the example above), the language specification says that

1. The contents of the string will be converted to a number, if possible; and
2. If that conversion was possible, then the resulting number will be divided by the other operand, the second number;
3. However, if the contents of the string cannot be converted to a number, the program will generate an error at runtime!

Think you can remember all that? Well, that's just the "easy" case: [The entire set of rules for converting between strings and numbers is much more extensive](#). In other words, as the author of the program, you have to remember all those subtleties while you are already trying to make code that works! Good luck! Without support from the compiler to detect these types of errors, a bug like this may lurk in the system for days, weeks, months or (even) years before causing our user trouble. After all that time, the code will be harder to read and understand and, therefore, the bug will be harder to fix!

In C++, a program with a type error will not even compile. That means there is no way that we programmers can overlook type errors and send a faulty program to our user.