Throwing Exceptions

The C++ exception handling feature allows programs to deal with real life circumstances. The system is broken into three parts:

- `try` blocks
- `catch` blocks
- `throw` statements

Let's start by looking at `throw` statements, so we can finish up our functions. Click on the running man to open the sample program we've been working on in **Replit**, **Fork it**, and we'll continue by learning how to apply exception handling.

When a function encounters a situation from which it cannot recover–for example, a call to `stoi("twelve")`—you can **report the error** by using the `throw` keyword to notify the nearest appropriate error-handling code that something has gone wrong.

```
istringstream in(str);
int result = 0;
if (in >> result) return result;
throw … // signal error at this point using throw
```

Inside `stoi()`, if the read succeeds, return the result. If the read **fails**, **jump out of the function**, looking for the closest error handler. You do that with `throw`.

Like `return`, `throw` accepts a single parameter, an object which provides information about the error which occurred.