Handling Errors

When you program, you’ll come across a lot of errors. And that’s ok! And when you’re first starting to learn a strongly-typed language, they can be pretty common.

So, what happens if you forget to specify a data type for your variable?:

```
randomData = "asdf jskdf";
Console.WriteLine(randomData);
```

C# will give an error, because it doesn’t want you to have random data being used in your application. The error will look something like this:

```
The name 'randomData' does not exist in the current context [CS0103:]
```

If you use the wrong type definition, like an `int` when it’s supposed to be a `double`:

```
int score = 45.39;
```

You might see an error like this:

```
Cannot implicitly convert type 'double' to 'int'. An explicit conversion
exists (are you missing a cast?)
```

We also want to watch out for how we name our variables. It’s good practice to use `camelCase` styling, and they should only contain underscores, letters, and digits.

```
string iAmAVariable;
string i'mnot; // this will cause errors
```

There are also a few words that you can’t use. These are known as *reserved keywords*. Reserved keywords are words that the language uses, so they already have specific definitions that shouldn’t be re-written. If we use one of them as a variable name, we risk overwriting it and causing significant errors in our program. For example, we can’t name a variable `string` because that word is reserved for defining data types.

Lastly, it’s important to double-check spelling and punctuation! Don’t forget to end each statement with a semicolon `;`.

Luckily, many IDEs will point out these potential errors before we even run our code! But it’s still good to be prepared to handle these errors if you should see them, including when you’re coding on Codecademy.

☑**Instructions**

1.

The following code has a couple of bugs in it. Run the code and see what errors appear. Once you know the errors, try fixing it line by line.

Hint

Here are the sources of errors:

- `number` has the wrong type.
- `dinosaur` is missing a type.
- `is.yes` is an illegal variable name.
- The `band` expression is missing a semicolon.