Testing Exceptions in JUnit

Testing that an exception is thrown when expected is an important type of unit test.

It works very similarly to our other tests in that some data setup is done, some action is performed, and an assert statement is run to see if the actual result meets our expected result.

The difference is in how we execute the code that could throw an exception, and then test to see whether that exception was thrown and if it was the expected exception.

Using assertThrows

The method to test if an expected exception was thrown is called `assertThrows` and it's part of the JUnit 5 package.

The first parameter you pass is the class of the exception you expect to be thrown. This is done in `Exception.class` format. For example: `NumberFormatException.class`

The second parameter is what's called a Java Lambda expression, and you may not have used one before. It is an inline function that you write and gets executed right there on the spot. The syntax is a little funny, but uses parenthesis to pass any needed outside arguments, an arrow, and then curly braces.

```
(arg) -> {System.out.println("I'm in a Lambda!");}
```

Inside the curly braces you can build out your function in as many lines of code as you need in normal Java syntax. You can put whatever you want in there: Loops, if statements, variable declarations, etc

When you put it all together in the assertThrows, it looks something like this:

```
@Test
    public void whenInputCostOfMaterialsIsString_thenThrowsException() {
        assertThrows(NumberFormatException.class, () -> {
            // Performing some data setup
            ByteArrayInputStream in = new
ByteArrayInputStream(JOB_NAME.getBytes());
            System.setIn(in);

            // Calling the code that will throw an exception with the data
scenario I set up above
            HarveyHomeRepair.inputCostOfMaterials();
        });
    }
```

If an exception is thrown that matches the one you placed as the first argument to `assertThrows` the assertion will be true and the test will pass. If an exception is not thrown, or one is thrown that does not match, the assertion will be false and the test will fail.