Return Errors

As we mentioned before, we don't like surprises — they lead to mistakes. So, when we call a method, we'd like to know what type of value will be returned. This is done in the method definition.

The method definition must contain the type of the return value: if a method returns an integer, its return type must be `int`; if it returns text, it must be `string`, and so on. If the method returns nothing, use `void`.

If a method returns a type different from its stated return type, it will throw an error. Here are some common errors you may see —

This error means you must state a return type before the method name:

```
error CS1520: Method must have a return type
```

This error means that your method doesn't return a value, when it should:

```
error CS0161: [MethodName]: not all code paths return a value
```

In some cases, this error means that your method returns a `string` when it should be an `int` (this one can be caused by a lot of things outside of methods):

```
error CS0029: Cannot implicitly convert type 'string' to 'int'
```

It's important to remember that running into errors is a natural part of coding. As a teacher once put it "Great programmers understand that errors are part of the process, and they know how to find the solution to each while learning something new from them."

☑Instructions

1.
This code has a bunch of errors! Run the code to find them.
2.

Fix the first error by adding a return type to one of the methods.

Hint

Fix the error by adding a return type to the `DecoratePlanets()` method.

3.

Fix the second error by adding a `return` to one of the methods.

Hint

Fix the error by adding a return statement to the `IsPlutoADwarf()` method.

4.

Fix the last error by changing the return type of one of the methods.

Hint

Fix the error by changing the return type of the `CountThePlanets()` method.