The Object Type

Every class is derived from `Object` — even classes that we define ourselves!

When you create a class, C# implicitly makes it inherit `Object`. So when we write this code:

```
class Book
{}
```

C# assumes we mean:

```
class Book : Object
{}
```

Even if we already declared a superclass, like...

```
class Book : Media
{}
```

... `Object` will be at the top of the family tree. Maybe `Media` directly inherits from `Object`, or its base class inherits from `Object`, etc.

We can prove this to ourselves by attempting to convert any reference to an `Object`. None of this code will throw an error because each type — at some point — inherits from `Object`:

```
Object o1 = new Dissertation();
Object o2 = new Diary();
Object o3 = new Random();
Object o4 = new Forest("Amazon");
```

Value types and strings also inherit from `Object`:

```
Object o5 = 21;
Object o6 = false;
Object o7 = "Hello you!";
```

☑Instructions

1.

There are three references created in **Program.cs**, one of type `Book`, one of type `Diary`, and one of type `int`.

Create a new `Object` reference for each one.

Hint

In other words, create three new variables of type `Object`, and set them equal to `bk`, `dy`, and `i`, respectively.

To pass this checkpoint you must declare and define the variable in the same line:

```
// This passes
Object obj = rand;

// This does not pass
Object obj;
obj = rand;
```