Public vs. Private

At this point we have built fields to associate data with a class and properties to control the getting and setting of each field. As it is now, any code outside of the `Forest` class can “sneak past” our properties by directly accessing the field:

```
f.Age = 32; // using property
f.age = -1; // using field
```

The second line avoids the property’s validation by directly accessing the field. We can fix this by using the *access modifiers* `public` and `private`:

> `public` — a public member can be accessed by any class
>
> `private` — a private member can only be accessed by code in the same class

For simplicity, we’ve been adding `public` to every member so far. That allows code to access the members from the `Main()` method, which doesn’t belong to the `Forest` class. When we switch a field from `public` to `private` it will no longer be accessible from `Main()`, although code inside the `Forest` class — like properties — can still access it.

Access modifiers can be applied to all members of a class, including fields, properties, and the rest of the members covered in this lesson.

Remember *encapsulation*? `public` and `private` are necessary to encapsulate our classes. Think of it like “defensive coding”: you are protecting the inner mechanisms of a class with `private` so that other code can’t break your class. You only expose what you want to be `public`.

For example, since a class’ properties define how other programs get and set its fields, it’s good practice to make fields private and properties public.

C# encourages encapsulation by defaulting class members to `private` and classes to `public`.

☑Instructions

1.

Currently the `biome` field and `Biome` property are public. In **Program.cs**, the field is directly accessed and set to `"Desert"`, an invalid value.

Run the code to see that “Desert” is the field’s value.

2.

By directly accessing the public `biome` field the code skipped the validation. Let’s prevent that by setting the `biome` field to private in **Forest.cs**.

When you run the code, you should see the error:

```
error CS0122: 'Forest.biome' is inaccessible due to its protection level
```

That means C# has prevented us from accessing a private field (which is good).

Hint

In **Forest.cs**, make sure to edit the field `biome` and not the property `Biome`.

3.

Address the error by changing the code in **Program.cs**: use the property (`Biome`) instead of the field (`biome`). You’ll need to change code in two places in the file.

What is printed to the console now?

Hint

After replacing `f.biome` with `f.Biome`, you should see “Unknown” printed to the console.