

# Static Fields and Properties

You already know how to create a field and property, like:

```
class Forest
{
    private string definition;
    public string Definition
    {
        get { return definition; }
        set { definition = value; }
    }
}
```

The definition of what a forest is applies to all `Forest` objects, not just one — there should only be one value for the whole class. This is a good use case for a static field/property.

To make a static field and property, just add `static` after the access modifier (`public` or `private`).

```
class Forest
{
    private static string definition;
    public static string Definition
    {
        get { return definition; }
        set { definition = value; }
    }
}
```

Remember that static means “associated with the class, not an instance”. Thus any static member is accessed from the class, not an instance:

```
static void Main(string[] args)
{
    Console.WriteLine(Forest.Definition);
}
```

If you tried to access a static member from an instance (like `f.Definition`) you would get an error like:

```
error CS0176: Static member 'Forest.Definition' cannot be accessed with an instance reference, qualify it with a type name instead
```

## ☒ Instructions

1.

In the previous exercise we mentioned storing the count of all `Forest` objects. We'll use a static field and property to store that. Define a private static field named `forestsCreated`.

Hint



`forestsCreated` should be an integer.

2.

Define a public static property named `ForestsCreated`. Give it a public getter and private setter.

Hint



Here's the first line of the property definition:

```
public static int ForestsCreated
```

The getter should be public and return `forestsCreated`.

The setter should be private and set `forestsCreated` to `value`.

3.

In the first constructor, increment `ForestsCreated` . This will add 1 to the property every time an object is constructed.

Hint



Within this constructor...

```
public Forest(string name, string biome)
```

...add 1 to `ForestsCreated` .