

[Return to Module Overview Page \(https://onlinelearning.berkeley.edu/courses/665040/pages/module-four\)](https://onlinelearning.berkeley.edu/courses/665040/pages/module-four)

## 4.4: Introduction to Access Control

So far, every definition we've written has started with the word *public*. Public member variables or public constructors are accessible to code outside the class in which they are defined; that means we can call the constructors and change the values of the variables whenever we want.

I hope a little bell has gone off in your head. Previously, we discussed the evils of exposing variables to the possibility of modification by code anywhere in a program, and how hiding or protecting them was a Good Thing. How is that relevant here? Suppose that (and I think this would be true in many applications) once a circle is created, it should be allowed to move from one position to another, but its size should remain fixed. Most real circle-shaped objects behave exactly this way. Pizzas (before you eat them!), hubcaps, and steering wheels move, for example, but their diameters do not change. We can enforce this in our code by making the radius *private*:

```
public class Circle
{
    public double x,y;
    private double r;
    // Create a generic Circle
    public Circle()
    {
        x = y = 0.0;
        r = 1.0;
    }
    // create a circle of a given size
    public Circle(double radius)
    {
        x = y = 0.0;
        r = radius;
    }
}
```

Now I can't change the size of a Circle once it's created:

```
// create a large circle
Circle large = new Circle(20.0);
//Compiler Error!
large.r = 1.0;
```

**Compiler Says:**

```
Circle.java:15: Variable r in class Circle not accessible from class main.
    large.r = 1.0;
        ^
```

Notice how code in class main cannot access the variable `r`, but code in `Circle` can. This means that we can point at all the code in the program that could possibly have changed private variables a huge help in debugging!

We've got a small problem now, however. Code in the class main can't change the size of a `Circle`, but neither can it find out what the size of a `Circle` is! We certainly don't want the size of a `Circle` to be a secret; it's part of the outward appearance of a real circle, after all. Therefore, we have to add a public function to the class:

```
public class Circle
{
    public double x,y;
    private double r;

    // Constructors as above

    // report the size of the Circle
    public double radius() { return r; }
}
```

The function `radius()` reports the radius of the circle, but does not allow the caller to change the radius. `radius()` is called a *member function*, a much more general concept than a constructor. A class can contain any number of public or private member functions, and the functions can accept any arguments and return any types they want.