# Access Control

**The goal for this exercise** is to understand the difference between public and private

Within the provided starter project, there is already a class named Access\_Control. You will add code to the RunExercise method so that you create two separate Blender objects. Each object has the speed that it’s currently operating at (which is a whole number from 0 (meaning ‘off’) to 7, inclusive at both ends)(inclusive means ‘including’ – inclusive at both ends means that both 0 (the low end) and 7 (the high end) are included in the range of valid values).

You should have a SetSpeed, GetSpeed, and Print methods. SetSpeed is the mutator/setter method (no values returned, pass in the new speed as the only parameter), and GetSpeed is the accessor/getter method (returns the current speed, accepts no parameters). Print will print out the current speed in the following format:

**Blender’s current speed:X**

Where X is the current speed of the blender. Note that this line should be followed by a new line (i.e., use Console.Write**Line**, not Console.Write, to print this out). The SetSpeed method should only change the value of the speed property if the provided parameter is valid (i.e., if it’s in the range of 0 through 7); otherwise nothing should happen.

What’s (slightly) different about this exercise is that you need to specify every instance variable, and every method, to be either public or private. A good rule of thumb is that everything should be private, unless you *specifically intend to use it in another class.* All your data fields (instance variables) should be private, unless you’ve got a really really good reason not to do so. (In C#, there are basically no good reasons – even if you thought you had a good reason, you can use something called a **property** to make it look like the instance variable is public, even when it’s not).

In this case, the purpose of SetSpeed/GetSpeed is specifically to allow another class to modify the speed setting on a Blender object. In this case, to allow the class Access\_Control (in it’s RunExercise method) to modify a Blender object, so making those public is reasonable. Similarly, the purpose of the method is to allow the class Access\_Control (in it’s RunExercise method) print out the current setting of a Blender, so making it public is reasonable, too.

The other thing that’s different is that you have to create the Blender class from scratch – put it where you normally do, just below the Access\_Control class. In Access\_Control.RunExercise, create (at least) two instances of the Blender class, and use the setter/getter and Print methods to change the speed on the Blender a couple times, printing out the current speed each time.

**What you need to do for this exercise:**

1. Implement the Blender class from scratch, including the SetSpeed, GetSpeed, and Print methods
2. Create at least two instances of the Blender class in Access\_Control.RunExercise, and use the Blender‘s methods.
3. Also, play around with this, and try accessing **private** members (try this for both private data & for private methods) in Access\_Control.RunExercise , in order to see what compile-time errors you get, etc, etc.

**As a general rule, from this point forwards, you should always make any data members in you classes (i.e., any instance variables) private, and use either accessor methods (like you've just done) or C# Properties (which you may see later on) to access the private data fields.**