

Students studying for the AP Computer Science exam, starting in the Fall of 2014, should append this text after the “Object Methods” section in Chapter Ten, Lesson Two.

Scope of Local Variables and Class Properties

We have demonstrated how to declare variables both locally (within one method) and as a class property. These two kinds of declarations have different lifespans and visibility that should be carefully understood. The concept of variable lifespan and visibility is called *scope*. A variable is "in scope" at a particular spot in your code if it exists and can be seen by code at that location. Otherwise, it is "out of scope".

All variables must be declared within some set of opening and closing curly braces. These braces will mark the beginning and ending of the variable's scope. If you declare the variable as a class property, then the variable will be visible to all of the methods within that class, and will live for as long as the class itself. But if you declare the variable within a method, then the variable will only exist within those curly braces and will disappear when the method exits. Consider this example:

```
class Car
{
    // this variable is declared within a class
    // but outside any function
    String color = "Blue";
    void start()
    {
        // myLocalInt is declared locally inside start()
        int myLocalInt = 3;
        // we can also use all class-level variables such as color
        System.out.println("My color is: " + color);
    }
    void stop()
    {
        // yes we can see class-level variables here too
        System.out.println("My color is: " + color);
        // ERROR: can't see local variables from other functions
        myLocalInt = 2;
    }
}
```

We have declared two variables: **color** is at *class-level* scope within the class curly braces but outside any method. This variable can be seen and used by all class methods and will live as long as the class does. When the class is destroyed, the class-level variable is also destroyed.

On the other hand, **myLocalInt** is declared within **start()** curly braces. This means the variable is *local* in scope to that method and will only live until the last statement in **start()** has finished. At that point the variable is destroyed, and it cannot be seen or used by any other method. Inside a method or function, you can only access variables that are locally declared within that method and variables that are declared at a class level within the object.