# Overloading methods: Creating Your Own

**The goal for this exercise** is to make sure that you can create your own, overloaded methods, so you can create your own, overloaded constructors in subsequent exercises.

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

1. Go back to your code from the prior exercise, and add an overloaded method to the Blender class.  
    Specifically, add a method named **Print(bool longWinded)** method on the Blender class that will print out a message like "**For this blender, the current speed is 3**", if **longWinded** is true. If it's false, it will print out something like " **Blender’s current speed:3**".   
   1. This should be in addition to the **Print()** method that you created for the previous exercise, which will print out the " **Blender’s current speed:3**" style of message to the user.
2. Make sure that you can call both methods from the RunExercise method in the Overloading\_Create\_Your\_Own class.
   1. If this method/class doesn't yet exist, you should create it on your own, and call the RunExercise method from main.
3. Also, experiment with this, so that you can figure out how to have one version of an overloaded method call another version of the same method. For example, since the 'concise' version of the overloaded version of Print prints the same thing as the version that has no parameters, instead of writing that code twice, have the parameterized version call the version that has no parameters.