Tracking Grades

A teacher wants a program to keep track of grades for students and decides to create a student class for his program as follows:

- Each student will be described by three pieces of data: his/her name, his/her score on test #1, and his/her score on test #2.
- There will be one constructor, which will have one argument—the name of the student.
- There will be three methods: *getName*, which will return the student's name; *inputGrades*, which will prompt for and read in the student's test grades; and *getAverage*, which will compute and return the student's average.

1. File *Student.java* contains an incomplete definition for the Student class. Save it to your directory and complete the class definition as follows:
   a. Declare the instance data (name, score for test1, and score for test2).
   b. Create a Scanner object for reading in the scores.
   c. Add the missing method headers.
   d. Add the missing method bodies.

2. File *Grades.java* contains a shell program that declares two Student objects. Save it to your directory and use the *inputGrades* method to read in each student's test scores, then use the *getAverage* method to find their average. Print the average with the student's name, e.g., "The average for Joe is 87." You can use the *getName* method to print the student's name.

3. Add statements to your Grades program that print the values of your Student variables directly, e.g.:

   ```
   System.out.println("Student 1: " + student1);
   ```

   This should compile, but notice what it does when you run it—nothing very useful! When an object is printed, Java looks for a *toString* method for that object. This method must have no parameters and must return a String. If such a method has been defined for this object, it is called and the string it returns is printed. Otherwise the default *toString* method, which is inherited from the Object class, is called; it simply returns a unique hexadecimal identifier for the object such as the ones you saw above.

   Add a toString method to your Student class that returns a string containing the student's name and test scores, e.g.:

   ```
   Name: Joe  Test1: 85  Test2: 91
   ```

   Note that the toString method does not call System.out.println—it just returns a string.

   Recompile your Student class and the Grades program (you shouldn't have to change the Grades program—you don't have to call toString explicitly). Now see what happens when you print a student object—much nicer!

Band Booster Class

In this exercise, you will write a class that models a band booster and use your class to update sales of band candy.

1. Write the BandBooster class assuming a band booster object is described by two pieces of instance data: *name* (a String) and *boxesSold* (an integer that represents the number of boxes of band candy the booster has sold in the band fundraiser). The class should have the following methods:
   - A constructor that has one parameter—a String containing the name of the band booster. The constructor should set boxesSold to 0.
   - A method *getName* that returns the name of the band booster (it has no parameters).
   - A method *updateSales* that takes a single integer parameter representing the number of additional boxes of candy sold. The method should add this number to *boxesSold*.
   - A *toString* method that returns a string containing the name of the band booster and the number of boxes of candy sold in a format similar to the following:

     Joe: 16 boxes

2. Write a program that uses BandBooster objects to track the sales of 2 band boosters over 3 weeks. Your program should do the following:
   - Read in the names of the two band boosters and construct an object for each.
   - Prompt for and read in the number of boxes sold by each booster for each of the three weeks. Your prompts should include the booster's name as stored in the BandBooster object. For example,

     Enter the number of boxes sold by Joe this week:

     For each member, after reading in the weekly sales, invoke the *updateSales* method to update the total sales by that member.
   - After reading the data, print the name and total sales for each member (you will implicitly use the toString method here).