Question 2: Step Tracker

Class: `StepTracker`	**9 points**

Intent: *Define implementation of a class to record fitness data*

+1 Declares all appropriate `private` instance variables

+2 Constructor

- **+1** Declares header: `public StepTracker(int ___)`
- **+1** Uses parameter and appropriate values to initialize instance variables

+3 `addDailySteps` method

- **+1** Declares header: `public void addDailySteps(int ___)`
- **+1** Identifies active days and increments count
- **+1** Updates other instance variables appropriately

+1 `activeDays` method

- **+1** Declares and implements `public int activeDays()`

+2 `averageSteps` method

- **+1** Declares header: `public double averageSteps()`
- **+1** Returns calculated `double` average number of steps

2019 SCORING GUIDELINES

Question 2: Scoring Notes

Class `StepTracker`			**9 points**
Points	Rubric Criteria	Responses earn the point even if they...	Responses will not earn the point if they...
+1	Declares all appropriate `private` instance variables		• omit keyword `private` • declare variables outside the class
+2	Constructor		
+1	Declares header: `public StepTracker(int ___)`	• omit keyword `public`	• declare method `private`
+1	Uses parameter and appropriate values to initialize instance variables	• initialize primitive instance variables to default values when declared	• fail to use the parameter to initialize some instance variable • fail to declare instance variables • initialize local variables instead of instance variables • assign variables to parameters
+3	`addDailySteps` method		
+1	Declares header: `public void addDailySteps(int ___)`	• omit keyword `public`	• declare method `private`
+1	Identifies active days and increments count	• put valid comparison erroneously in some other method	• fail to use the parameter as part of the comparison • fail to increment a count of active days • fail to increment an instance variable • compare parameter to some numeric constant
+1	Updates other instance variables appropriately		• update another instance variable only on active days • update another instance variable inappropriately • fail to update appropriate instance variable • update a local variable
+1	`activeDays` method		
+1	Declares and implements `public int activeDays()`	• return appropriate count of active days where the instance variables were updated improperly in `addDailySteps` or `activeDays`	• declare method `private` • return value that is not the number of active days • fail to return a value

2019 SCORING GUIDELINES

Question 2: Scoring Notes (continued)

Points	Rubric Criteria	Responses earn the point even if they...	Responses will not earn the point if they...
+2	`averageSteps` method		
+1	Declares header: `public double averageSteps()`	• omit keyword `public`	• declare method `private`
+1	Returns calculated `double` average number of steps	• maintain instance variables improperly but calculate appropriate average • fail to handle the special case where no days are tracked	• use integer division • calculate something other than steps divided by days • fail to return

2019 SCORING GUIDELINES

Question 2: Step Tracker

```
public class StepTracker
{
   private int minSteps;
   private int totalSteps;
   private int numDays;
   private int numActiveDays;

   public StepTracker(int threshold)
   {
      minSteps = threshold;
      totalSteps = 0;
      numDays = 0;
      numActiveDays = 0;
   }

   public void addDailySteps(int steps)
   {
      totalSteps += steps;
      numDays++;
      if (steps >= minSteps)
      {
         numActiveDays++;
      }
   }

   public int activeDays()
   {
      return numActiveDays;
   }

   public double averageSteps()
   {
      if (numDays == 0)
      {
         return 0.0;
      }
      else
      {
         return (double) totalSteps / numDays;
      }
   }
}
```

These canonical solutions serve an expository role, depicting general approaches to solution. Each reflects only one instance from the infinite set of valid solutions. The solutions are presented in a coding style chosen to enhance readability and facilitate understanding.