Honor Code: All submissions should include a comment statement near the top of the program of the form:

```java
/* THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING
 * A TUTOR OR CODE WRITTEN BY OTHER STUDENTS - YOUR NAME
 */
```

Cases of apparent plagiarism or collusion will be referred to the Honor Council.

Preparation: To disallow other students from reading your homework programs, you must save your file(s) in a directory inside your cs170 directory. If you follow the below commands, your work will be protected.

1. Create a directory called hw9 inside your cs170 project directory to save your hw9 files.
   ```bash
   mkdir ~/cs170/hw9
   ```

2. You must use ~/cs170/hw9 directory as your current directory when editing any program files for hw9. Change your current directory to your newly created hw9 directory:
   ```bash
   cd ~/cs170/hw9
   ```

3. You can now run gedit to edit your programs:
   ```bash
   gedit yourProgramName.java &
   ```
   The name yourProgramName is the name of the Java program (and also the name of the class!).

1. (100 pts)
   - Write a Java class named Appointment.java representing appointments you make. You will be modeling the equivalent of a real world appointment using Java.

   - This class should include several pieces of data as instance variables:
     - day of the week: (integer with 0 representing Sunday and 6 representing Saturday)
     - who the appointment is with (String)
     - duration of the appointment (double)
     - start time of the appointment (double with 0.0 representing midnight, 12.0 representing noon, and (for example) 15.0 representing 3pm. For this assignment, you can assume all appointments begin either on the hour or on the half-hour.)

   - The Appointment class will have several methods
     - get/set methods for each instance variable (all of which should be declared private).
       ```java
getDay, setDay
```
getPerson, setPerson
getDuration, setDuration
getStartTime, setStartTime

• 3 constructors: a default constructor with no parameters, a constructor with 4 parameters (day, start time, duration, and person in that order), and a copy constructor, which makes a new object from the current state of another object (passed as a parameter).

• the toString method. As discussed in class, this method is very useful for working with (and debugging!) objects and object variables. It should return a reasonable representation of the object based on the instance variable values, but the exact format of that String is up to you.

• the method reschedule which takes a new start time and a new duration and updates the current appointment.

• the method conflictsWith which takes an Appointment object as a parameter. This function returns a boolean value based on whether or not the current appointment and the parameter variable appointment conflict with each other. For the purposes of this assignment, a conflict is defined as two appointments on the same day of the week whose times overlap by some amount.

Testing your solution:

• You will need to write a class, say AppointmentTests.java, which instantiates (creates) a number of Appointment object variables and tests your functions above. How you write this file is entirely up to you, as it is strictly for your own testing purposes.

Submission:

• Submit your work using the following commands. You need to be in your ~/cs170/hw9 directory when you issue them.
  ◦ /home/cs170001/turnin Appointment.java hw9
• Your homework is not turned unless the above commands are successful (you will get a "success" message when turn in was successful).