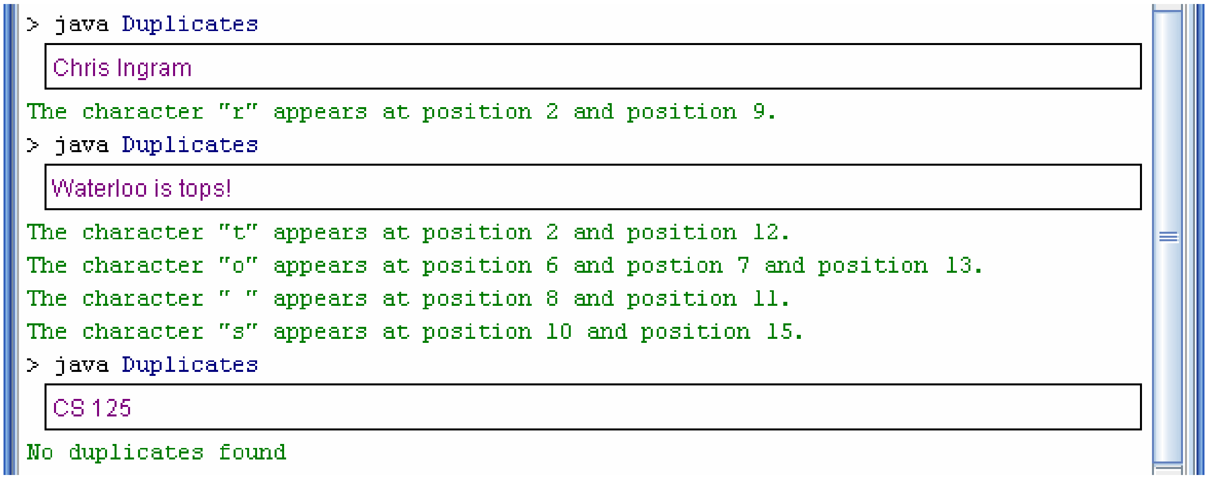
Question 1: Duplicate Counter (30%)

Write a program called `Duplicates.java` which will determine whether a string contains duplicate characters, and where they occur.

Your program should read in a single line of input. For each character in the string starting from the left, if that character appears again later in the string your program should print out all the positions that the character appears.

If every character in the string is unique, your program should display an appropriate message. Do not print out the duplicate locations for the second and later occurrences of the character.

The following is the output from three separate runnings of the program:



```
> java Duplicates
  Chris Ingram
The character "r" appears at position 2 and position 9.
> java Duplicates
  Waterloo is tops!
The character "t" appears at position 2 and position 12.
The character "o" appears at position 6 and postion 7 and position 13.
The character " " appears at position 8 and position 11.
The character "s" appears at position 10 and position 15.
> java Duplicates
  CS 125
No duplicates found
```

Note:

- In the first example the capital and lower case “I” are considered different characters.
- In the second example there is only one line of output for the character “o”.

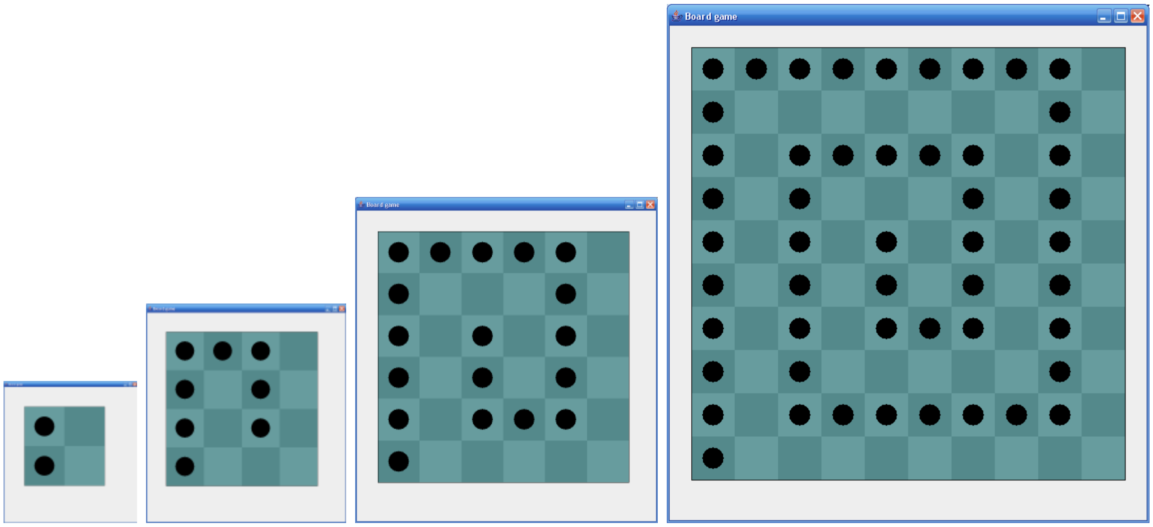
Files to Submit: `Duplicates.java`

Question 2: Spiral Pattern (30%)

Write a program called `Spiral.java` which will draw a spiral pattern onto a square board using black pegs.

Your program should read in a number which you may assume is even and positive. After creating a square board of that size, draw the spiral pattern (starting from the bottom left corner of the board working towards the center) as indicated in the diagrams below.

Here is output for 2, 4, 6 and 10 squares. Note that the squares without pegs should form the exact same pattern as the squares with pegs, but rotated 180 degrees.



As a first step, you should try to reproduce the spiral using the `drawLine` method (this can be accomplished using a single loop that draws four line segments at a time). Then you need only replace each `drawLine` by a loop which puts a line of pegs on the board. For half credit you can submit your solution that uses `drawLines` instead of placing pegs.

Files to Submit: `Spiral.java`

Question 3: Rate Your Professor (30%)

For the next exercise, you are to write a class called `Professor.java` which specifies how to make an object that maintains information about a teacher similar to what you would find on professor rating sites (e.g. www.ratemyprofessors.com).

When the instructor is created someone provides their name and department.

At this point students can submit a rating of their abilities. A rating consists of:

- A ranking from 1-5 in three categories (easiness, helpfulness and clarity). Rankings that are outside the range are given the closest valid ranking.
- Whether the student believes they are attractive or not (hotness)
- A comment they want others to be able to read. A comment is cut off after 40 characters.

At any point in time, someone can find out the following information about the teacher:

- The name and department of the instructor
- The number of times the instructor has been rated
- The average easiness rating the instructor was given (0.0 if unrated)
- The overall quality of the instructor which is the average of all the helpfulness and quality ratings they received (0.0 if they are unrated)
- The instructor's smiley (emoticon) rating. If they are at least 3.5 it is "**:)** ", if they are unrated or at least 2.5 it is "**:**|". All others are "**:(**".
- Whether people think the instructor is "hot". If more people rated the instructor as attractive than those that did not, they are considered "hot".
- The two most recent comments (most recent given first).

We have provided a template of the class in the starter file. Read the documentation carefully. You cannot change any of the method signatures provided, or add new public methods. You will decide what instance variables you need to keep track of, although we have given you some suggestions. Note, you are creating a class, not an actual program. None of the methods should produce any output.

Files to Submit: `Professor.java`

Question 4: Ratings Program (10%)

Complete the program called `Ratings.java` which will help you test your solution to question 3.

The program will ask the user for the names and departments of two professors. Then it will create two `Professor` objects. The program will proceed to loop doing the following:

- Asks the user which professor to enter a rating for next. A number other than 1-2 signifies the program should end.
- Reads in three numbers from the user corresponding to the Easy, Helpful and Clarity rankings.
- Reads in a string corresponding to whether they think the instructor is attractive. Any input other than "no" should be treated as the person thinking they are attractive.
- Reads in a line of input corresponding to their comment.
- Submits a rating for that professor.
- Prints out an updated "report card" for that professor. The report card contains:
    - On line 1 - The smiley, name, department and number of times rated
    - On line 2 – The overall quality, average easy rating and whether they are "hot"
    - On the next line(s) - The most recent two comments

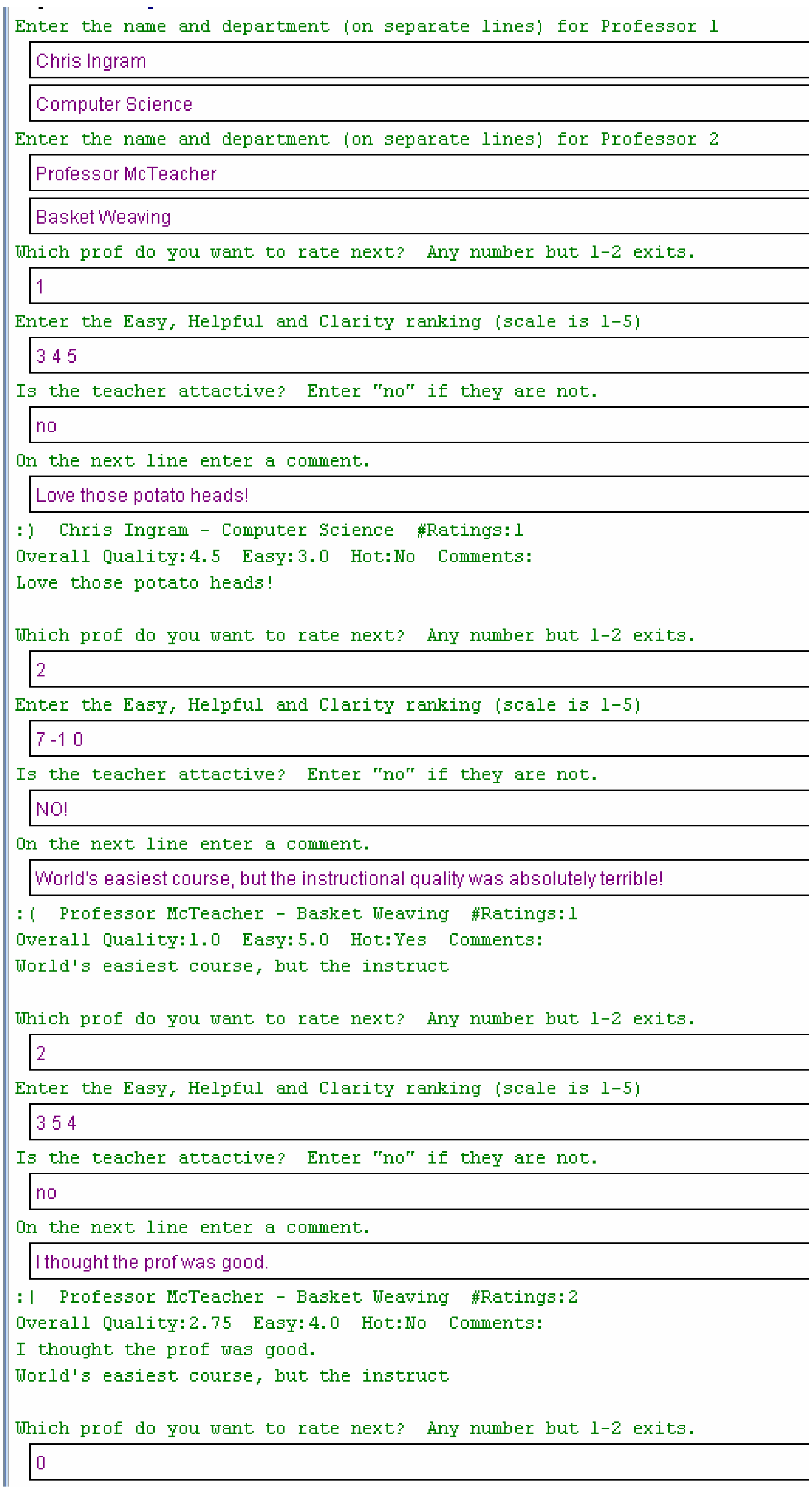
The next page contains a sample run from the ratings program.

The provided file already handles prompting the user for input, and reading it in. Your task is to add code in the indicated locations of this file to call the appropriate methods on the objects to complete the program and produce output that matches the sample run.

This is primarily an exercise in calling the correct Professor methods, and building the appropriate strings to generate the missing report card output.

You may assume the user always enters numbers when they are supposed to.

Files to Submit: `Ratings.java`



```
Enter the name and department (on separate lines) for Professor 1
Chris Ingram
Computer Science
Enter the name and department (on separate lines) for Professor 2
Professor McTeacher
Basket Weaving
Which prof do you want to rate next?  Any number but 1-2 exits.
1
Enter the Easy, Helpful and Clarity ranking (scale is 1-5)
3 4 5
Is the teacher attactive?  Enter "no" if they are not.
no
On the next line enter a comment.
Love those potato heads!
:)  Chris Ingram - Computer Science  #Ratings:1
Overall Quality:4.5  Easy:3.0  Hot:No  Comments:
Love those potato heads!

Which prof do you want to rate next?  Any number but 1-2 exits.
2
Enter the Easy, Helpful and Clarity ranking (scale is 1-5)
7 -1 0
Is the teacher attactive?  Enter "no" if they are not.
NO!
On the next line enter a comment.
World's easiest course, but the instructional quality was absolutely terrible!
:(  Professor McTeacher - Basket Weaving  #Ratings:1
Overall Quality:1.0  Easy:5.0  Hot:Yes  Comments:
World's easiest course, but the instruct

Which prof do you want to rate next?  Any number but 1-2 exits.
2
Enter the Easy, Helpful and Clarity ranking (scale is 1-5)
3 5 4
Is the teacher attactive?  Enter "no" if they are not.
no
On the next line enter a comment.
I thought the prof was good.
:|  Professor McTeacher - Basket Weaving  #Ratings:2
Overall Quality:2.75  Easy:4.0  Hot:No  Comments:
I thought the prof was good.
World's easiest course, but the instruct

Which prof do you want to rate next?  Any number but 1-2 exits.
0
```