NAME: login ID:

precept:

COS 126 Midterm 2 Programming Exam, Spring 2008

This part of the exam is like a mini-programming assignment. You will create a program **with comments**, compile it, create test data for it and run it on your laptop. Debug it as needed. Your program will be graded on style. This exam is open book, open browser. You may use code from your assignments or code found on the COS126 website. When you are done, submit it via the course website using the Social Activities link called Precept Exam 2.

Put your name, login ID, and precept number on this page (*now*), and write out and sign the Honor Code pledge before turning in this paper. Note: It is a violation of the Honor Code to discuss this midterm exam question with anyone until after everyone in the class has taken the exam. You have 50 minutes to complete the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

_____

Signature

1 /25

Description of Program (25 points) – Overlaps version

In one dimension, a line segment is an ordered pair of numbers that defines a set of points on the line. Your task is to develop a Java class `Segment` that implements this abstraction and allow clients to test whether two line segments overlap. Specifically, implement the following API:

```
public class Segment
        Segment (double x0, double x1)
boolean contains(double x)          is x in this line segment?
boolean overlaps(Segment that)      do this segment and that overlap?
String toString()                   convert segment to the format  x0 -- x1
```

You should consider line segments to exclude the endpoint `x0` but include endpoint `x1.` The arguments to the constructor are ordered, so that `Segment(x0, x1)` represents all points *x* on the line in the half-closed interval `(x0, x1]`.

Organize your program as a class `Segment` with a `main()` test client that takes an integer `N` as command-line argument, reads `N` pairs of `double` values from standard input, makes an array of `Segment` objects, uses `overlaps()` to find the intersecting pairs, and prints out all the overlapping pairs. You will need to have `StdIn` in your directory in order to compile and test your program.

For example, if the file `tiny.txt` contains

```
1.2 2.5
2.0 5.0
8.1 9.7
6.5 7.7
3.4 8.0
```

then `java Segment 5 < tiny.txt` should give the following results:

```
1.2 -- 2.5 overlaps 2.0 -- 5.0
2.0 -- 5.0 overlaps 3.4 -- 8.0
6.5 -- 7.7 overlaps 3.4 -- 8.0
```

Your output need not be in this order, but it must contain all of these lines.

Remember to submit your program `Segment.java` on the course website.

Description of Program (25 points) – No overlap version

In one dimension, a line segment is an ordered pair of numbers that defines a set of points on the line. Your task is to develop a Java class `Segment` that implements this abstraction and allow clients to test whether two line segments overlap. Specifically, implement the following API:

`public class Segment`	
`Segment (double x0, double x1)`	
`boolean contains(double x)`	*is x in this segment?*
`boolean overlaps(Segment that)`	*do this segment and that overlap?*
`String toString()`	*convert segment to the format* `x0 -- x1`

You should consider line segments to include the endpoint `x0` but exclude endpoint `x1.` The arguments to the constructor are ordered, so that `Segment(x0, x1)` represents all points *x* on the line in the half-closed interval `[x0, x1)`.

Organize your program as a class `Segment` with a `main()` test client that takes an integer `N` as command-line argument, reads `N` pairs of `double` values from standard input, makes an array of `Segment` objects, uses `overlaps()` to find all line segments that overlap *no* others, and prints out all such line segments. You will need to have `StdIn` in your directory in order to compile and test your program.

For example, if the file `tiny.txt` contains

```
1.2 2.5
2.0 5.0
8.1 9.7
6.5 7.7
7.8 8.0
```

then `java Segment 5 < tiny.txt` should give the following results:

```
8.1 -- 9.7
6.5 -- 7.7
7.8 -- 8.0
```

Your output need not be in this order, but it must contain all segments that overlap no others.

Remember to submit your program `Segment.java` on the course website.