____________ Seat Number Name ________________________________

CS 1713 Exam 3 — Spring 2012

This is a closed book exam. Answer all questions on these sheets. You may use a calculator, but not one on a cell phone or other connected device. Please put your name and seat number of the first and last page.

1) (15 points) Write a method to sort an array of **`double`** in increasing order using an insertion sort. You may (and it is recommended that you) have your sort method call another method, but you need to write that other method also.

2) (15 points)

   a) Write a public method, **`linearSearch`**, that uses a linear search to find the first position in an **`ArrayList`** of **`String`** of a given **`String`**. Return -1 if it is not found.

   b) Write a public method, **`binarySearch`**, that uses a binary search to find a position in an **`ArrayList`** of **`String`** of a given **`String`**. Return -1 if it is not found. Assume the **`ArrayList`** is sorted in increasing order.

3) (12 points) Note: a millisecond is .001 seconds.

   1) It takes about 5 milliseconds to search a given array of length 200,000 using a linear search. How many **seconds** would you expect it to take to search a similar array of of size 6,000,000? Show how you got your answer.

   b) It takes about 100 milliseconds to sort a given array of size 200,000 using a selection sort. How many **seconds** would you expect it to take for the selection sort to sort a similar array of size 6,000,000? Show how you got your answer.

   c) It takes about 2 milliseconds to search a sorted array of 2 million elements using a binary search. How long would you expect it to take to search a similar sorted array of 16 million elements? Show how you got your answer.

4) (6 points) The **`MyDate`** class has month, a day, and a year, all of which are integers. We wish to have this class implement **`Comparable`** so that dates can be sorted chronologically. Assume reasonable names for the attributes and write a **`compareTo`** method for this class.

5) (11 points) Write the method:

```
public static String processInfo(String s);
```

It is expected that the parameter consists of a name followed by a colon (:), followed by a single blank and an integer. The name may contain any characters other than a colon. If the parameter has the expected form and the integer is greater than 0, the returned **`String`** should consist of the integer followed by a blank and the name. If the parameter does not have the expected format, the returned **`String`** should consist of the **`String`** `"Invalid: "` followed by the original parameter.

Examples:

parameter	return value	comment
`Robbins: 37`	`37 Robbins`	OK
`Robbins 37`	`Invalid: Robbins 37`	no colon
`Robbins:`	`Invalid: Robbins:`	no integer
`Robbins: -3`	`Invalid: Robbins: -3`	negative integer
`Robbins: a12`	`Invalid: Robbins: a12`	invalid integer
`Robbins: 37 xyz`	`Invalid: Robbins: 37 xyz`	additional data after the integer
`Robbins:  37`	`Invalid: Robbins:  37`	extra blank between colon and integer

6) (11 points) Write a method:

```
public static boolean processFile(String infile, String outfile);
```

It will create the file `outfile` from the file `infile`. For each line of `infile` call the method `processInfo` from the previous problem to produce the corresponding line in `outfile`. Return true if there are no errors reading or creating the files, and return false if any error occurs. This method should not do any printing. Use the following for input and output:

```
new Scanner(new FileInputStream(filename))
new PrintWriter(new PrintWriter(filename),true))
```

7) (10 points) Write a method that takes two parameters, and array of **`Rectangle`** and a **`double`**. It returns the width of the first **`Rectangle`** in the array whose area is greater than the second parameter. Return -1 if there is no such **`Rectangle`**. You may assume that the **`Rectangle`** class has already been written and has appropriate methods.

8) (4 points) Draw an accurate schematic diagram of the program variables showing the execution of the program.

```
int x = 5;
int y = 9;
String s = "ABCD";
String t = "XYZ";
String u = s;
t = s.substring(2);
x = s.indexOf("C");
y = t.indexOf("C");
```

9) (6 points) Suppose **`Circle`** is a class that has been appropriately written. It has a constructor with one **`double`** parameter, its radius. Draw an accurate schematic diagram of the program variables showing the execution of the program.

```
double x = 2;
double y = 3;
double z = 4;
Circle[] circles = new Circle[4];
Circle c;
c = new Circle(x);
circles[0] = new Circle(y);
circles[1] = c;
circles[2] = circles[0];
circles[3] = circles[1];
circles[0].setRadius(6);
z = circles[2].getRadius();
```

10) (10 points) The tables below show an array of 8 integers. Fill in each table with a trace of the indicated sort applied to this array. The sort will sort the elements in ascending (increasing) order. Each line in the table should be filled in with the values of the array after one pass through the main loop of the sort. There may be are more lines in the table than needed.

Selection Sort

`list[0]`	`list[1]`	`list[2]`	`list[3]`	`list[4]`	`list[5]`	`list[6]`	`list[7]`
71	24	14	62	87	75	10	31

Insertion Sort

`list[0]`	`list[1]`	`list[2]`	`list[3]`	`list[4]`	`list[5]`	`list[6]`	`list[7]`
71	24	14	62	87	75	10	31

____________ Seat Number Name ________________________________