

CS 1713 Exam 3 Solutions — Spring 2012

1) (15 points)

```
public static void insertionSort (double [] list) {
    for (int index = 1; index < list.length; index++)
        insertItem(list, index);
}

public static void insertItem(double [] list, int index) {
    double key = list[index];
    int position = index;

    while ((position > 0) && (key < list[position-1])) {
        list[position] = list[position-1];
        position--;
    }
    list[position] = key;
}
```

2a) (4 points)

```
public int linearSearch(ArrayList<String> array, String target) {
    for (int i=0; i<array.size(); i++)
        if (array.get(i).equals(target))
            return i;
    return -1;
}
```

2b) (11 points)

```
public int binarySearch(ArrayList<String> array, String target) {
    int low = 0;
    int high = array.size() - 1;
    while (low <= high) {
        int mid = (low + high)/2;
        if (array.get(mid).equals(target))
            return mid;
        else if (array.get(mid).compareTo(target) < 0)
            low = mid + 1;
        else
            high = mid - 1;
    }
    return -1;
}
```

3) (4 points each) a) $30 \times 5\text{ms} = 150\text{ ms} = .15\text{ seconds}$ b) $30 \times 30 \times 100\text{ ms.} = 90\text{ seconds.}$ c) $21/24 = 2\text{ms}/x$, so $x = 2.28\text{ms.}$

4) (6 points)

```
public int compareTo(Object other) {
    MyDate otherDate = (MyDate)other;
    if (year != otherDate.year)
        return year - otherDate.year;
    if (month != otherDate.month)
        return month - otherDate.month;
    return day - otherDate.day;
}
```

5) (11 points)

```
public static String processInfo(String s) {
    int ind;
    String invalidString = "Invalid: "+s;
    String name;
    String valString;
    ind = s.indexOf(": ");
    if (ind == -1)
        return invalidString;
    name = s.substring(0, ind);
    valString = s.substring(ind+2);
    try {
        ind = Integer.parseInt(valString);
        return ind + " " + name;
    } catch (Exception e) {
        return invalidString;
    }
}
```

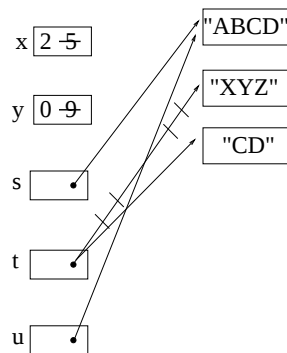
6) (11 points)

```
public static boolean processFile(String infile, String outfile) {
    Scanner scan = null;
    PrintWriter out = null;
    String line;
    try {
        scan = new Scanner(new FileInputStream(infile));
        out = new PrintWriter(new PrintWriter(outfile), true);
        while (scan.hasNextLine()) {
            line = scan.nextLine();
            out.println(processInfo(line));
        }
        return true;
    } catch (Exception e) {
        return false;
    } finally {
        scan.close();
        out.close();
    }
}
```

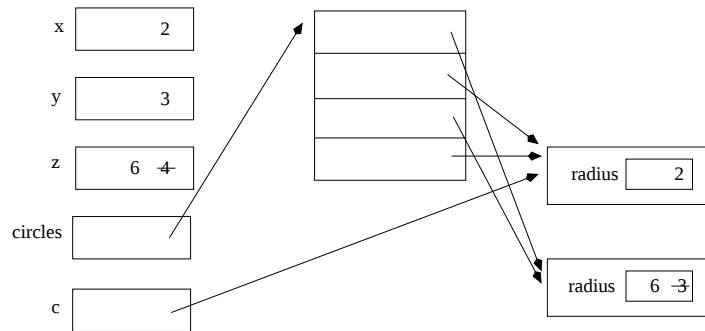
7) (10 points)

```
public static double widthOfLargeRectangle(Rectangle[] list, double minArea) {
    for (int i=0; i<list.length; i++)
        if (list[i].getArea() > minArea)
            return list[i].getWidth();
    return -1;
}
```

8) (4 points)



9) (6 points)



10) (10 points)

Selection Sort

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

Insertion Sort

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