CSC 1351: Exam 2: Midterm

VERSION B

1 Sorting

1.1

Fill in the missing code. The number of lines corresponds to the answer key. Your code may vary.

```
public class Cat implements Comparable {
  int numKittens; // sort by this if the names are equal
  String name; // sort by this first
  @Override
  public int compareTo(Object o) {
    Cat that = (Cat)o;
    ..........
    int diff = this.name.compareTo(that.name);
    ..........
    if(diff != 0) return diff;
    ..........
    return this.numKittens - that.numKittens;
    ..........
  }
  public static void main(String[] args) {}
}
```

1.2

What kind of sort is this? Rewrite it to use ArrayList.

```
  static void sort(int end,int[] values) {
    if(end < 2)
      return;
    for(int i=1;i<end;i++) {
      if(values[i-1] > values[i]) {
        int tmp = values[i];
        values[i] = values[i-1];
        values[i-1] = tmp;
      }
    }
    sort(end-1,values);
  }
```

This is a bubble sort because it's constantly comparing adjacent elements.

```
  static void sort(int end,ArrayList<Integer> values) {
    if(end < 2)
      return;
    for(int i=1;i<end;i++) {
      if(values.get(i-1) > values.get(i)) {
        int tmp = values.get(i);
        values.set(i,values.get(i-1));
        values.set(i-1,tmp);
      }
    }
    sort(end-1,values);
  }
```

2 Anonymous Inner Classes

2.1

Given the interface defined like this:

```
public interface MouseListener {
  void mousePressed(int x,int y);
  void mouseReleased(int x,int y);
}
```

Write a complete program that uses an anonymous inner class that implements this interface, then calls mousePressed() to print the x and y values of the mouse event to the screen.

```
public class Test {
  public static void main(String[] args) {
    MouseListener ml = new MouseListener() {
      public void mousePressed(int x,int y) {
        System.out.println("("+x+","+y+")");
      }
      public void mouseReleased(int x,int y) {
      }
    };
    ml.mousePressed(230,127);
  }
}
```

3 Recursion

3.1

You want to implement flipping a coin on a computer. However, you want to be fancy and include the possibility of the coin standing on edge, with a probability of 2%. In that case you flip it again, until the coin lands on either side. "Heads" give 1 point, "Tails" give 0 points, and an edge gives an additional 2 points (potentially multiple times). Most of the time the return value will be 0 or 1. However, if the coin landed on edge once, the return value will be either 2 or 3, depending on the final toss. If it landed on edge twice, it could be either 4 or 5, in a similar way. Fill in the missing code. The number of lines corresponds to the answer key. Your code may vary.

```
  final static Random rand = new Random();
  public static int flip() {
    int randomPercentage = rand.nextInt(100);
    if (randomPercentage <  2) return 2 + flip();
    ..........
    if (randomPercentage < 51) return 0;
    ..........
    return 1;
    ..........
  }
```

3.2

What is wrong with the following program, what happens when you run it? If this method would have been implemented recursively, and would have a similar error, what would happen then if run?

```
public class JavaIsToJavascriptWhatCarIsToCarpet {
  public static int factorial(int n) {
    int result = 1;
    while (n > 1) {
      result *= n;
    }
    return result;
  }
```

```
  public static void main(String args[]) {
    assert(factorial(4) == 24);
  }
}
```

There is an exit-condition missing in the while loop. When run, the program hangs. This is similar to infinite recursions, so the same in a recursive program would result in a stack overflow.

4 Big O

4.1

What is the Big-O notation for the following function?
$T(N) = 100000 + 10N - 3 + N^2$
The leading term as N gets large is N^2, so...
$O(N^2)$

4.2

What is the Big O speed for a linear search? For a merge sort? For an insertion sort?
For linear search it's $O(N)$.
For merge sort it's $O(N \log N)$
For insertion sort it's $O(N^2)$