SAMPLE EXAM
Final Exam
Computer Programming 326
Dr. St. John
Lehman College
City University of New York
Thursday, 16 December 2010

NAME (Printed)	______________________
NAME (Signed)	______________________
E-mail	______________________

Exam Rules

- Show all your work. Your grade will be based on the work shown.
- The exam is closed book and closed notes.
- When taking the exam, you may have with you pens or pencils, and an 8 1/2" x 11" piece of paper filled with notes, programs, etc.
- You may not use a computer or calculator.
- All books and bags must be left at the front of the classroom during this exam.
- **Do not open this exams until instructed to do so.**

Question 1	
Question 2	
Question 3	
Question 4	
Question 5	
Question 6	
Question 7	
Question 8	
Question 9	
Question 10	
TOTAL	

1. True or False:
   (a) ___ Arrays can only be of primitive type.
   (b) ___ You can pass an entire array to a method.
   (c) ___ Within a constructor, `super` calls a constructor of the same class.
   (d) ___ In Java, every class is a descendant of the predefined class `Object`.
   (e) ___ An exception is caught in a `try` block.
   (f) ___ Each `try` block can have multiple `catch` blocks.
   (g) ___ All non-text files are referred to as binary files.
   (h) ___ Closing a file disconnects it from a stream.
   (i) ___ Every recursive method can be rewritten as an iterative method.
   (j) ___ Every iterative method can be rewritten as a recursive method.

2. (a) What is an exception? Give an example:

   (b) How does an exception differ from an error?

3. (a) Write the first line of a java class called `SchoolKid` that extends the class `Kid`:

   (b) Write the first line of a java class called `Square` than implements the interface `Measurable`:

   (c) Write the first line of a java class called `MyApplet` that extends the interface `JApplet` and implements `ActionListener`:

4. What is the output when the code is run?

(a) **Output:**

```
char[] s = {'h', 'i', 'm', 'o', 'm'}
for (int j=0; j < s.length-1; j++) {
   if ( s[j]  > s [j+1] ) {
      System.out.print(" +");
   }
   else
     System.out.print("  -");
}
System.out.println();
}
```

(b) **Output:**

```
int[] A = {5, 4, 3, 2, 1};
for (int i = 0; i < A.length; i++)
    System.out.print(A[i] + " ");
System.out.println();
for (int i = 0; index < A.length - 1; i++) {
   int smallest= getIndexOfSmallest(i, A);
   System.out.println("Exchanging: " + A[i] +
        " " + A[smallest]);
   interchange(i, smallest, A);
}
for (int i = 0; i < A.length; i++)
    System.out.print(A[i] + " ");
```

Assuming the following method definitions:

```
private static int getIndexOfSmallest(int startIndex, int[] a) {
    int min = a[startIndex];
    int indexOfMin = startIndex;
    for (int index = startIndex + 1; index < a.length; index++) {
       if (a[index] < min) {
          min = a[index];
          indexOfMin = index;
       }
    }
    return indexOfMin;
}
private static void interchange(int i, int j, int[] a)
{
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp; //original value of a[i]
}
```

5. Find at least 5 of the errors in the following *and explain* how to fix them:

```
public static void sort(int[] a) {
  int i, j
  for (i = 0; i > a.length-1; a++)
    for (j = 0; j < a.length-1; i++) {
      if ( a[j] = a[j+1] ) {
        a[j] = a[j+1];
        a[j+1] = a[j];
      }
    }
}
```

6. Given the method:

```
public static void mystery(String begin, String end) {
    if (endingString.length() <= 1)
       System.out.println(begin + end);
    else
      for (int i = 0; i < end.length(); i++) {
        try {
           String newString = end.substring(0, i) + end.substring(i + 1);
           mystery(begin + endi.charAt(i), newString); }
         catch (StringIndexOutOfBoundsException exception) {
           exception.printStackTrace(); }
     }
}
```

What is the output from the following statements?

(a) `System.out.print(mystery("","I"));`

Output:

(b) `System.out.print(mystery("by","I"));`

Output:

(c) `System.out.print(mystery("", "ab"));`

Output:

(d) `System.out.print(mystery("a","b"));`

Output:

(e) `System.out.print(mystery("","abc"));`

Output:

7. Given the classes below, indicate the number of possible outputs, as well as the output itself:

```
public static class Thread1 extends Thread {
  public void run() {
    System.out.println("A");
  }
}
public static class Thread2 extends Thread {
  public void run() {
    System.out.println("1");
    System.out.println("2");
    System.out.println("3");
  }
}
```

(a) **Number of Outputs:** **Outputs:**

```
new Thread1().start();
```

(b) **Number of Outputs:** **Outputs:**

```
new Thread1().start();
new Thread1().start();
```

(c) **Number of Outputs:** **Outputs:**

```
new Thread1().start();
new Thread2().start();
```

(d) **Number of Outputs:** **Outputs:**

```
new Thread2().start();
new Thread1().start();
```

(e) **Number of Outputs:** **Outputs:**

```
new Thread1().start();
new Thread1().start();
new Thread1().start();
```

8. Given the following interface:

```
/**
 An interface for methods that return
 the perimeter and area of an object.
*/
public interface Measurable
{
    /** Returns the perimeter. */
    public double getPerimeter();

    /** Returns the area. */
    public double getArea();
}
```

Implement a class `Circle` that implements the interface. It should include a constructor, instance variables, and definitions for methods in the interface for a circle.

9. (a) Write a method that takes as input a file stream and a maximum length, `m`, and returns the first `m` characters of the next line if it exists. If the file is empty, the method should return the empty string, "".

```
public static String getLineLength( InputStream in, int m ) {








}
```

(b) Using the method above, write a `paint()` method for a `JApplet` that writes the next line of the stream `data` to the graphical user interface. If there is no next line, you should display: "`No more lines`."

```
public static void paint() {








}
```

10. Write a **complete** program that sorts a list of numbers. You may use any sort you wish. Print out the list of numbers before and after you sort.