

MACS 261J
Final Exam
May 7, 2007

Name: _____

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Points:	5	7	5	8	15	5	5	25	5	10	10	100
Score:												

Question 1 (5 points)

Rename the following method so that it is more descriptive. Using javadoc comments, write the documentation for this method. Include descriptions of method parameters and the returned value.

```
public static int someMethod(float x, float[] a) {
    int n = a.length;
    for (int i=0; i<n; ++i)
        if (x==a[i])
            return i;
    return -1;
}
```

Question 2 (7 points)

(a) [5 points] What does this program fragment print?

```
System.out.print("I like ");
for (int n=946; n>0; n=n/10)
    System.out.print(n%10);
System.out.printf(" but square root of 2 is %1.2f.%n",Math.sqrt(2));
```

(b) [2 points] What happens if you change $n>0$ to $n>=0$?

Question 3 (5 points)

Complete the following method so that it prints Dave's triangle (not as famous as Pascal's triangle) for any integer value n . For example, for $n=5$ Dave's triangle is:

```
0
1 2
2 3 4
3 4 5 6
4 5 6 7 8
```

Your implementation should work for any positive n .

```
public static void printDavesTriangle(int n) {

}
}
```

Question 4 (8 points)

Suppose you see the following two statements in someone else's program.

```
float y = Math.sin(Math.sqrt(3.937));
double x = 4.0*Math.atan(1.0);
```

- (a) [2 points] The first statement has an error and will not compile. Fix it.

- (b) [2 points] The second statement computes a well-known value. Simplify it.

- (c) [2 points] How would you enable the class name `Math` to be omitted?

- (d) [2 points] Write the declaration (not implementation) of the method `Math.sqrt`. In other words, in the class `Math`, how is this method declared?

Question 5 (15 points)

Complete the following methods:

(a) [5 points] (array of arrays)

```
/**
 * Returns a 2-D array of integers corresponding to a chess board.
 * (An alternating 8 x 8 grid of black and white squares.)
 * For each black square, the board contains the integer value 0.
 * For each white square, the board contains the integer value 1.
 * The square corresponding to array index [0][0] is black.
 * @return array of array of ints representing the chess board.
 */
public static int[] [] makeChessBoard() {

}
}
```

(b) [5 points] (indexing array elements forwards and backwards)

```
/**
 * Returns a new array with elements reversed from the specified array.
 * @param x array from which to copy reversed elements; not modified.
 * @return the new array.
 */
public static float[] reverse(float[] x) {

}
}
```

(c) [5 points] (swapping array elements, two at a time)

```
/**
 * Reverses elements of the specified array, in place.
 * @param x array with elements to be reversed.
 */
public static void reverseInPlace(float[] x) {

}
}
```

Question 6 (5 points)

Consider the method `foo` defined by:

```
public static int foo(int i, int j, int k) {
    if (i<j) {
        if (j<k) {
            return j;
        } else if (i<k) {
            return k;
        } else {
            return i;
        }
    } else if (j>k) {
        return j;
    } else if (i>k) {
        return k;
    } else {
        return i;
    }
}
```

(a) [3 points] What does the following program fragment print?

```
System.out.println(foo(1,2,3));
System.out.println(foo(2,1,3));
System.out.println(foo(3,2,1));
```

(b) [2 points] Rename the method `foo` to better describe what it does.

Question 7 (5 points)

Give examples that illustrate proper use of the operators `=` and `==` and the standard method `equals(Object that)` shared by all objects.

Question 8 (25 points)

Implement the class Car by filling in the blanks below.

```
public class Car {
    /**
     * Constructs a car with specified parameters.
     * @param milesPerGallon miles the car can go for each gallon of gas.
     * @param tankCapacity maximum number of gallons that tank can hold.
     */
    public Car(double milesPerGallon, double tankCapacity) {

    }

    /**
     * Turns this car 90 degrees right.
     */
    public void turnRight() {

    }

    /**
     * Turns this car 90 degrees left.
     */
    public void turnLeft() {

    }

    /**
     * Adds the specified number of gallons of gas to this car's tank.
     * @param gallons number of gallons to add.
     * @throws BadEnvironmentException if the tank overflows.
     */
    public void addFuel(double gallons) {

    }
}
```

```
/**
 * Tries to drive this car the specified distance in the current direction.
 * Goes only as far as possible given the gas currently in this car's tank.
 * @param distance the number of miles to go.
 * @return the number of miles actually gone.
 */
public double go(double distance) {

}

/**
 * Returns the number of gallons of gas remaining in this car's tank.
 * @return the number of gallons.
 */
public double readFuelGauge() {

}

/**
 * Gets the distance from this car to the point where it was constructed.
 * Note that this distance does not equal the total number of miles driven.
 * This method assumes that the car travels in a plane and returns the
 * distance between two points in that plane.
 * @return the distance, in miles.
 */
public double getDistance() {

}
```

```
// Declare private fields here.  
// Hint: use short field names,  
// like mpg for milesPerGallon.
```

```
} // end of class Car
```

Question 9 (5 points)

Write a subclass `BadEnvironmentException` of `java.lang.RuntimeException`. In your subclass include one constructor with a `String` message parameter that is simply passed to the corresponding constructor of the superclass.

Question 10 (10 points)

Complete the following program to construct a new `Car` and then call methods to add fuel, go 30 miles, turn right, then go 40 miles, and finally to print the distance from the starting point (not the total distance travelled). Do not run out of gas.

```
public class RoadTrip {  
    public static void main(String[] args) {
```

```
    }  
}
```

Question 11 (10 points)

Write a complete Java program with a method `main` that (1) asks the user to enter a number `x`, and then (2) computes and prints the cosine of `x`. In your program, do something reasonable in the *exceptional* case where the user enters some text that does not represent a number.

(Hint: this is the only question in this exam for which your answer should include use of the standard class `java.util.Scanner`.)