Solutions

1. (20 Points) Complete the following Java program by filling in the bodies of functions sum(), avg() and ord(). A call to sum(n) should return the sum of all the integers from 1 to n (inclusive), while avg(n) returns the average of the same set of numbers. A call to the boolean function ord(x, y, z) returns true if \( x < y < z \) and false otherwise. Function main() should produce the following output:

```
6.5
true
false
```

// Problem1.java
class Problem1{

    // sum(): return 1+2+3+..+n
    static int sum(int n){
        int s=0;
        for(int i=1; i<=n; i++){
            s += i;
        }
        return s;
    }

    // avg(): return average of {1,2,..,n} (Hint: use function sum())
    static double avg(int n){
        return sum(n)/(double)n;
    }

    // ord(): return true if and only if x<y<z
    static boolean ord(double x, double y, double z){
        return x<y && y<z;
    }

    public static void main(String[] args){
        System.out.println(avg(12));
        System.out.println(ord(1.2, 3.4, 5.6));
        System.out.println(ord(3.4, 1.2, 5.6));
    }
}

2. (20 Points) Determine the output of the following Java program and place that output on the lines below exactly as it would appear in the terminal window. More lines are provided than are actually needed. Assume the program is invoked with the command line argument "blah" (i.e. `java Problem2 blah`), and that there exists a file in the current working directory called "blah" containing the single line "3 4". (That's the number 3, followed by a space, followed by the number 4).

```java
// Problem2.java
import java.util.Scanner;
import java.io.*;
class Problem2{
    public static void main(String[] args) throws FileNotFoundException{
        Scanner sc = new Scanner(new File(args[0]));
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(foo(a+b));
    }
    static int foo(int n){
        System.out.println( "in foo("+n")" );
        if(n>0){
            return foo(n-1) + 6;
        }else{
            return 3;
        }
    }
}
```

**Program Output:**
```
in foo(7)
in foo(6)
in foo(5)
in foo(4)
in foo(3)
in foo(2)
in foo(1)
in foo(0)
45
```
3. (20 Points) Fill in the definitions of functions Transpose() and printArray() in the following Java program. Function Transpose() will compute and return the transpose of its 2-dimensional String array argument. In other words, Transpose() will create a new 2-dimensional String array object whose rows are the columns if its argument X[][], and whose columns are the rows of X[][] . Function printArray() will print out the elements of its array argument one row per line, so that the output of function main() is as follows.

```
abc def ghi
klm nop qrs

abc klm
def nop
ghi qrs
```

// Problem3.java
class Problem3{
    public static void main(String[] args){
        String[][] A = {{"abc", "def", "ghi"}, {"klm", "nop", "qrs"}};
        String[][] B = Transpose(A);
        printArray(A);
        System.out.println();
        printArray(B);
    }

    static String[][] Transpose(String[][] X){
        String[][] T = new String[X[0].length][X.length];
        for(int i=0; i<T.length; i++){
            for(int j=0; j<T[0].length; j++){
                T[i][j] = X[j][i];
            }
        }
        return T;
    }

    static void printArray(String[][] X){
        for(int i=0; i<X.length; i++){
            for(int j=0; j<X[0].length; j++){
                System.out.print(X[i][j]+" ");
            }
        }
        System.out.println();
    }
}
4. (20 Points) Complete the definition of function `join()` in the Java program below. This function takes as input three `int` arrays `A[]`, `B[]` and `C[]`, of any length, and returns a new `int` array containing the concatenation of `A[]`, `B[]` and `C[]`. If `join()` is written correctly, the output of function `main()` will be the single line: `1 2 3 4 5 6 7 8 9`.

```java
// Problem4.java
class Problem4{

    public static void main(String[] args){
        int[] X = {1, 2, 3, 4};
        int[] Y = {5, 6, 7};
        int[] Z = {8, 9};
        int[] J = join(X, Y, Z);

        for(int i=0; i<J.length; i++){
            System.out.print(J[i]+" ");
        }
        System.out.println();
    }

    static int[] join(int[] A, int[] B, int[] C){
        int[] D = new int[A.length + B.length + C.length];
        int i;

        // copy A[] into D[]
        for(i=0; i<A.length; i++){
            D[i] = A[i];
        }

        // copy B[] into D[]
        for(i=0; i<B.length; i++){
            D[A.length + i] = B[i];
        }

        // copy C[] into D[]
        for(i=0; i<C.length; i++){
            D[A.length+B.length+i] = C[i];
        }

        return D;
    }
}
```
5. (20 Points) Determine the output of the following Java program and place that output on the lines below exactly as it would appear in the terminal window. Again more lines are provided than necessary. This program consists of two files: Animal.java which defines the Animal class, and AnimalTest.java which contains function main().

// Animal.java
class Animal{
    // Fields
    String type;
    String name;

    // Method
    void printAnimal(){
        System.out.println(name+" is a "+type+".");
    }
}

// AnimalTest.java
class AnimalTest{
    public static void main( String[] args ){
        Animal A = new Animal();
        Animal B = new Animal();
        Animal C = new Animal();
        Animal temp;

        A.type = "dog";
        A.name = "Luke";
        B.type = "cat";
        B.name = "Chester";
        C.type = "Turtle";
        C.name = "Churchy";

        if( A.name.compareTo(B.name)>0 ) {temp = A; A = B; B = temp;}
        if( B.name.compareTo(C.name)>0 ) {temp = B; B = C; C = temp;}
        if( A.name.compareTo(B.name)>0 ) {temp = A; A = B; B = temp;}

        A.printAnimal();
        B.printAnimal();
        C.printAnimal();
    }
}

Program Output:
Chester is a cat.
Churchy is a Turtle.
Luke is a dog.