Solutions

1. (20 Points) Determine the output of the following Java program. Print the output on the lines provided below exactly as it would appear in the terminal window. Note that more lines are provided than are actually needed. Notice also that function $f()$ is overloaded, so there are actually three different functions called $f()$ in this program.

```java
// Problem1.java
class Problem1{
    public static void main(String[] args){
        int a=3, b=2, c;
        double x=5.0, y=4.0, z;

        c = f(a, b);
        b = f(c, x);
        z = f(x, y);
        System.out.println("a=", a, ", b=", b, ", c=", c);
        System.out.println("x=", x, ", y=", y, ", z=", z);
    }

    static int f(int i, int j){
        int k = (i++) + (++j);
        System.out.println("in f(int, int)");
        return 3*k;
    }

    static int f(int n, double t){
        System.out.println("in f(int, double)");
        return n*(int)t;
    }

    static double f(double u, double v){
        System.out.println("in f(double, double)");
        return (double)f((int)(2*u-v), 2.0);
    }
}

Program Output:
in f(int, int)
in f(int, double)
in f(double, double)
in f(int, double)
a=3, b=90, c=18
x=5.0, y=4.0, z=12.0
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. Again 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 "2 4". (That's the number "2", followed by a space, followed by the number "4").

// 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("in foo:  n = " + n);
    }
}

static int foo(int n){
    System.out.println("in foo:  n = " + n);
    if(n>1){
        return foo(n-1) + 8;
    }else{
        return 5;
    }
}

Program Output:
in foo:  n = 6
in foo:  n = 5
in foo:  n = 4
in foo:  n = 3
in foo:  n = 2
in foo:  n = 1
45
3. (20 Points) Determine the output of the following Java program, placing the output on the lines below exactly as it would appear on the terminal window. (Hint: first figure out what each of the methods \( f() \), \( g() \) and \( h() \) do, then put the pieces together.)

```java
// Problem3.java
class Problem3{
    public static void main(String[] args){
        int[] A = {2, 1, -5};
        int[] B = {4, -1, 3};
        int[][] T = g(A, B);
        System.out.println(f(A, B));
        h(T);
    }
    static int f(int[] X, int[] Y){
        int i, sum = 0;
        for(i=0; i<X.length; i++){
            sum += X[i]*Y[i];
        }
        return sum;
    }
    static int[][] g(int[] X, int[] Y){
        int[][] M = new int[X.length][Y.length];
        for(int i=0; i<X.length; i++)
            for(int j=0; j<Y.length; j++)
                M[i][j] = X[i]*Y[j];
        return M;
    }
    static void h(int[][] M){
        for(int i=0; i<M.length; i++){
            for(int j=0; j<M[i].length; j++)
                System.out.print(M[i][j]+" ");
        }
        System.out.println();
    }
}
```
(20 Points) Fill in the definition of the static function `concatenate()` in class Problem4 below. This function takes two int arrays `A` and `B` of any length as input, and returns a new int array which is the concatenation of its two input arrays. If `concatenate()` is written correctly, the output of function `main()` will be the single line: 1 2 3 4 5 6 7 8

```java
// Problem4.java
class Problem4{
    public static void main(String[] args){
        int[] X = {1, 2, 3, 4, 5};
        int[] Y = {6, 7, 8};
        int[] Z = concatenate(X, Y);

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

    static int[] concatenate(int[] A, int[] B){
        // your code starts here

        int[] C = new int[A.length + B.length];
        int i;

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

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

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

        /* yet another way to copy B[] into C[]
        for(int j=0; j<B.length; j++){
            C[i] = B[j];
            i++;
        }
        */

        return C;

        // your code ends here
    }
}
```
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: Point.java which defines the Point class, and PointTest.java which defines the PointTest class and contains function main().

// Point.java
class Point{
    // Fields
    int xcoord;
    int ycoord;
    // Constructor
    Point(int x, int y){ xcoord = x; ycoord = y;}
    // Methods
    public String toString(){ return "(" + xcoord + ", " + ycoord + ")"; }
    void swap(){ int temp = xcoord; xcoord = ycoord; ycoord = temp;}
    boolean isRightOf(Point P){ return (this.xcoord > P.xcoord);}
    boolean isAbove(Point P){ return (this.ycoord > P.ycoord);}
}

// PointTest.java
class PointTest{
    public static void main( String[] args ){
        Point A = new Point(2, 5);
        Point B = new Point(1, 3);
        Point C = new Point(7, -3);
        Point D = new Point(8, 4);
        String str1, str2;
        B.swap();
        C.swap();
        System.out.println("A = " + A);
        System.out.println("B = " + B);
        System.out.println("C = " + C);
        System.out.println("D = " + D);
        str1 = A.isRightOf(B)?"right":"left";
        str2 = C.isAbove(D)?"above":"below";
        System.out.println("A is to the " + str1 + ", " + str1 + " of B");
        System.out.println("C is " + str2 + " D");
    }
}

Program Output:
A = (2, 5)
B = (3, 1)
C = (-3, 7)
D = (8, 4)
A is to the left of B
C is above D