

CMPS 11

Introduction to Programming

Midterm 1 Review Problems

Note: The necessary material for some of these problems may not have been covered by end of class on Monday, the lecture before the exam. If so, those problems should be considered as review for midterm 2.

1. Determine the output of the following Java program. Assume that the user enters the numbers 5 3 9 7 0 on a single line, followed by return.

```
// Problem1.java
import java.util.Scanner;
class Problem1{
    public static void main( String[] args ){
        Scanner sc = new Scanner(System.in);
        int i, x, y, a, c=0, s=0;
        i = sc.nextInt();
        x = y = i;
        while( i!=0 ){
            if( i<x ) x = i;
            if( i>y ) y = i;
            s += i;
            c++;
            i = sc.nextInt();
        }
        a = s/c;
        System.out.println(x);
        System.out.println(y);
        System.out.println(a);
    }
}
```

2. Write a complete syntactically correct java program that prompts the user for two double values x and y , then prints out the value of the expression $\sqrt{x} + \sqrt{y}$. Include all necessary import statements, a class definition and `main()` function. You may give the class any valid name. Specify the name of the file that contains your program in a one-line comment at the beginning of the program.
3. The following Java program contains 3 syntax errors. (a) Circle each of the three errors and indicate how to correct them. (b) Determine the output of the program after the errors are corrected. (c) Re-write the corrected program by changing the while loop to a for loop.

```
// Problem3.java
class Problem3{
    public static void main(string[] args){
        int i = 1, p = 1;
        while( i<=5 ){
            p *= i,
            I++;
        }
        System.out.println(p);
    }
}
```

4. Determine the output of the following Java program.

```
// Problem4.java

class Problem4{

    public static void main( String[] args ){

        int a=2, b=5, c;
        double x=2.5, y=3.5, z;
        c = f(a, b);
        z = g(x, c);
        a = h(y, z, x);
        x = f( (int)z, b);
        System.out.println(a + " " + b + " " + c );
        System.out.println(x + " " + y + " " + z );

    }

    static int f(int n, int m){
        int k = n + m;
        k *= 2;
        return k;
    }

    static double g(double s, int n){
        return n*s;
    }

    static int h(double r, double s, double t){
        int m = (int)(r + s + t);
        return f(m, 1);
    }

}
```

5. 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`, 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
```

```
// Problem5.java
class Problem5{

    // sum(): return 1+2+3+...+n
    static int sum(int n){

    }

    // avg(): return average of {1,2,...,n}
    static double avg(int n){

    }

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

6. Trace the following Java program and place the output on the lines below *exactly* as it would appear on the screen. Assume that the user enters 1 3 5 at the first prompt, and 1.5 2.0 2.5 at the second prompt. (More lines printed below than are actually needed.)

```
// Problem6.java
import java.util.Scanner;
class Problem6{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int a, b, c;
        double x, y, z;
        System.out.print("Enter three integers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        c = sc.nextInt();
        System.out.print("Enter three doubles: ");
        x = sc.nextDouble();
        y = sc.nextDouble();
        z = sc.nextDouble();
        a = (++c)-(b++);
        x += (z - y);
        b *= b;
        y = c*z;
        c = (int)(b - x);
        System.out.println("a=" + a + ", b=" + b + ", c=" + c);
        System.out.println("x=" + x + ", y=" + y + ", z=" + z);
        System.out.println("c/a=" + c/a);
        System.out.println("c%a=" + c%a);
    }
}
```

7. Determine the output of the following Java program for any and all possible inputs by the user. In other words, describe the program's operation with sufficient generality so that, given any sequence of key presses by the user, one could predict the printed result.

```
// Problem7.java
import java.util.Scanner;

class Problem7{

    public static void main( String[] args ){

        Scanner sc = new Scanner(System.in);
        int a = 0;
        double x=0, y=0,z=0;

        while( a<30 ){

            while(true){

                while( !sc.hasNextDouble() ){
                    sc.next();
                }

                if( a<10 ){
                    x = sc.nextDouble();
                    if( x>0 ) break;
                }else if( a<20 ){
                    y = sc.nextDouble();
                    if( y<0 ) break;
                }else{
                    z = sc.nextDouble();
                    if( z>0 ) break;
                }
            }
            a += 10;
        }
        System.out.println("x = " + x + ", y = " + y + ", z = " + z);
    }
}
```

8. Write a complete Java program that reads 5 integers entered by the user, and prints them out in increasing order. Your program should continually prompt for integer input, discarding tokens that are not integers, until 5 integers have been entered by the user. Hint: study the `CheckInput` sequence of examples and `CollatzProblem.java` to see how to filter user input, and emulate the examples `Sort3.java` and `Sort4.java` to sort the 5 integers in increasing order.

9. Write a static method called `square()` that will take an integer and return its square and a static method called `cube()` that will take an integer and return its cube. Call your `square()` and `cube()` methods from within two new methods you write called `quartic()` and `quintic()` that return the fourth and fifth power of an integer, respectively. Use these methods to write a program that prints a table of powers of integers from 1 to 25. The output of your program should look like:

Integer	Square	Cube	Quartic	Quintic
1	1	1	1	1
2	4	8	16	32
.
.
.
25	625	15625	390625	9765625

10. Determine what is printed by the following code fragment.

```
int i=0, j=1, a=2, b=3, c=4;
a += b/a;
if(a<b)
    c += (c<a+b ? ++i : j++);
else
    c *= (c-a>b ? j-- : --i);
System.out.println("i="+i+", j="+j+", a="+a+", b="+b+", c="+c);
```