CSCI 201.001 Exam 2 Fall 2007 Solution

16 October, 2007

Problem 1 (12 points)

What is the difference between a **formal** and **actual** parameter?

The formal parameters are given in the method definition. For example, in

```
    public static int sillyMethod(int x)
```

seen in Problem 2, the formal parameter is `x`.

The actual parameters are the values passed to the method when called. For example, in

```
    sillyMethod(3)
```

seen in Problem 2, the actual parameter is `3`.

Does changing the name of either the formal or actual parameter change the signature of a method? Justify your answer.

In Java, the signature of a method is the it name and the types (not names) of its parameters. For example, in

```
    public static sillyMethod(int x)
```

the method's signature would be something like

```
    sillyMethod(int)
```

This all you need to know about a method to call it. The names of the formal parameters are not part of the method signature.

Problem 2 (12 points)

What is printed when the `main` method of the following pointless Java program is executed? If you desire partial credit you should explain the reasoning behind your answer.

```
package quiz2prob4;

public class Main {

    public static int sillyMethod(int x) {
        return (x+1)/2 ;
    }

    public static double sillyMethod(double x) {
        return x/2 ;
    }

    public static void main(String[] args) {
        System.out.println("First is " + sillyMethod(3)) ;
        System.out.println("Last is " + sillyMethod(4.0)) ;
    }
}
```

```
First is 2
Last is 2.0
```

Problem 3 (16 points)
In the boxes below are four method headers taken directly from the documentation of the Java `Math` class. For each, give a one-line example of legally calling each method and storing its returned value in an appropriately typed Java variable.

`static int abs(int a)`
`int k = Math.abs(-5) ;`

`static int getExponent(double d)`
`int k = Math.getExponent(5.0) ;`

`static double pow(double a, double b)`
`double y = Math.pow(2.0, 0.5) ;`

`static double scalb(double d, int scaleFactor)`
`double y = Math.scalb(2007.5, 3) ;`

Problem 4 (12 points)
Neither of the Java methods shown below are legal. For each, point out the problem.

```
public static int prob5a(int k) {
     for (int i=0; i<k; ++k) {
          System.out.println('*') ;
     }
     for (int i=0; i<k; ++k) {
          System.out.println('+') ;
     }
     System.out.println(i) ;
     System.out.println(k) ;
}
```

In the statement
`System.out.println(i)`
`i` is outside the scope of either of its previous definitions. The loop will only terminate if k is negative, but that's not the reason why this method will not compile.

```
public static int prob5b(int k) {
     for (int i=0; i<k; ++k) {
          for(int k=0; k<5; ++k) {
               System.out.println(k) ;
          }
          System.out.println(i) ;
     }
     System.out.println(k) ;
}
```

In the for-loop, the initialization
`int k=0`
is an attempt to redeclare `k` within a scope where it is already active.

Problem 5 (12 points)
What do each of the two following programs print when given the input sequence

```
201
202
203
```

```
input java.util.Scanner ;

static public void Main(String[] args) {
  Scanner stdin = new Scanner(System.in) ;
  for (int i=200; i<202; ++i) {
    int j = stdin.nextInt() ;
    System.out.println(j+1) ;
  }
}
```

```
202
203
```

```
input java.util.Scanner ;

static public void Main(String[] args) {
  Scanner stdin = new Scanner(System.in) ;
  int j = stdin.nextInt() ;
  while (j<202) {
    System.out.println(j+1) ;
    j = stdin.nextInt() ;
  }
}
```

```
202
```

Problem 6 (36 points)
Complete the following method so that it produces output like the following:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

In the above case, the method was passed the value that is assigned to `n`. You should write your program so that it produces `n` lines, each with `n` numbers. If it doesn't have a nested loop, it probably isn't correct.

Start with the following header

```
public static void prob7(int n) {
    for (int i=1; i<=n; ++i) {
        for (int j=1; j<=i; ++j) {
            System.out.print(j + ” ”) ;
        }
        System.out.println() ;
    }
}
```