

**CS 212 – Object-Oriented Programming in Java – Spring 2012 – Final Exam**  
**SOLUTIONS**

Last Name \_\_\_\_\_ First Name \_\_\_\_\_ SEAT \_\_\_\_\_

**Directions:** There are five questions with varying point values. **Read the whole question before answering.** Proper use of Java concepts is expected; minor syntax errors will be overlooked.

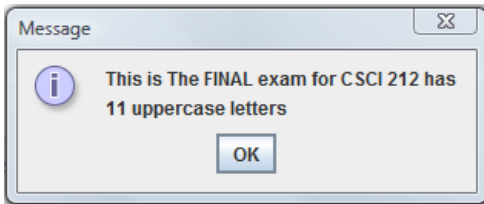
**Question 1. (20 points)**

Write a Java application that will take one argument from the **command line**, and write output to a *JOptionPane* message dialog saying how many upper case letters appeared in the command line argument.

For example,

```
java Question1 "This is The FINAL exam for CSCI 212"
```

Should produce:



If there is no command line argument an *IllegalArgumentException* should be thrown.  
The boolean method *isUpperCase* in the *Character* wrapper class may be of use.

```
import javax.swing.JOptionPane; // or javax.swing.*;
public class Question1 {
    public static void main (String[] args) {

        public static void main(String[] args) {
            if (args.length==0)
                throw new IllegalArgumentException ("Must have command line argument");
            int upperCount = 0;
            String input= args[0];
            for (int i=0; i<input.length(); i++){
                if (Character.isUpperCase(input.charAt(i)))
                    upperCount++;
            }
            JOptionPane.showMessageDialog(null, input+" has \n"+upperCount+" uppercase letters");
        }
    }
}
```

1\_\_\_\_/20

2\_\_\_\_/50

3\_\_\_\_/20

4\_\_\_\_/5

5\_\_\_\_/5

Question 2 refers to the Java program on the last two pages of the exam. You may remove the last two pages if it makes it easier to answer the questions on the exam.

**Question 2a. (10 points)**

What will be the printed output of the application FinalMainA.java when it is compiled and run?

```
Hello!
Hello!  p is 5
Let's keep going...
```

**Question 2b. (10 points)**

What will be the printed output of the application FinalMainC.java when it is compiled and run?

```
Hello!
2 plus 3 equals 5
2+3+4=9
Hello!  p is 10
```

**Question 2c. (10 points)**

What will be the printed output from the method *sub1* of the application FinalMainB.java when it is compiled and run?

```
p is 2
hello!  p is 6
hello! hello! Hello!
hello! hello! Hello!  p is 8
```

**Question 2d. (10 points)**

What will be the printed output from the method *sub2* of the application FinalMainB.java when it is compiled and run?

```
hello! hello! Hello! hello! Hello!
2 plus 5 equals 7
7
5 plus -3 equals 2
2
-3 plus 7 equals 4
4
```

**Question 2e. (10 points)**

What will be the printed output from the method *sub3* of the application FinalMainB.java when it is compiled and run?

```
hello! hello! Hello! hello! Hello! Hello!
10 plus 20 equals 30
10+20+20=50
50
```

**Question 3. (20 points)**

Assume a linked list is defined as follows. Write the code for the method *equals* in class *FinalList*. Two lists are equal if they have the same number of nodes and the corresponding nodes on each list are equal.

```
public class FinalListNode {
    public Object data;
    public FinalListNode next;
    public FinalListNode () {
        data=null;
        next=null;
    }
    public boolean equals (FinalListNode other) {
        return (data.equals(other.data));
    }
}

public class FinalList {
    private FinalListNode first, last;
    private int length;
    public FinalList() {
        first = new FinalListNode();
        last = first;
        length = 0;
    } // FinalList

    public boolean equals (FinalList other) {

        if (other == null || length != other.length) return false;

        FinalListNode p,q;
        p=this.first.next;
        q=other.first.next;

        while (p!=null && q!=null) { // && q!=null is not really necessary
            if (!p.data.equals(q.data))
                return false;
            else {
                p=p.next;
                q=q.next;
            }
        }
        return true;
    }
}
```

**Question 4. (5 points)**

Using the definitions of the `FinalListNode` and the `FinalList` as shown in Question 3, write a recursive static method to be placed in the class `FinalList` to determine if the two lists are equal.

It would be called as follows, assuming *myListA* and *myListB* have been created as *FinalLists*:

We assume here that *first* and *next* are public.

```
boolean theyAreEqual = equal (myListA.first.next, myListB.first.next);

public static boolean equal(FinalListNode theList, FinalListNode theOtherList) {

    if (theList==null && theOtherList==null) return true;
    if (theList==null && theOtherList != null) return false;
    if (theList!=null && theOtherList ==null) return false;
    if (!theList.equals(theOtherList)) return false;
    return (equal(theList.next,theOtherList.next));

}
```

**Question 5. (5 points)**

Suppose a Queens College course is written as a 2-5 capital letter department (e.g., CSCI, RM, MUSIC), a space, a 1-3 digit course number followed by an optional “W” (writing intensive). Create a Regular Expression that would recognize such courses.

Some examples of courses are

```
CSCI 212
CMLIT 101W
CHEM 16
RM 701
```

\_\_\_\_\_ `[A-Z]{2,5}\s\d{1,3}W?` \_\_\_\_\_

You may remove these two pages to make it more convenient to answer the questions.

```
public class FinalClassA {
    static String hello="";
    protected int p;
    private int q;

    public FinalClassA(){
        hello += "Hello! ";
        System.out.println(hello);
    }
    public FinalClassA(int c) {
        if (c<0) throw new FinalClassAException("C can't be negative");
        p = c;
        System.out.println(hello+" p is "+p );
        hello+="hello! ";
    }
    public int sum (int a, int b) {
        System.out.println(a+" plus "+b+" equals "+(a+b));
        return a+b;
    }
    public int times (int a, int b) {
        System.out.println(a+" times "+b+" equals "+(a*b));
        return a*b;
    }
}

public class FinalClassB extends FinalClassA implements FinalThings {
    public FinalClassB () {
        super();
    }
    public FinalClassB (String c) {
        hello+=c;
    }
    public FinalClassB (int c) {
        super(c+1);
    }
    public int sum (int a, int b, int c) {
        int answer = sum(a,b)+c;
        System.out.println(a+"+"+b+"+"+c+"="+answer);
        return answer;
    }
    public int times (int a, int b) {
        return a*b;
    }
    public int times (int a, int b, int c) {
        int answer = times(a,b)*c;
        System.out.println(a+"*"+b+"*"+c+"="+answer);
        return answer;
    }
}

public interface FinalThings {
    int c=0;
    public int sum (int a, int b);
    public int times (int a, int b);
    public int times (int a, int b, int c);
}
```

```
public class FinalMainA {
    public static void main(String[] args) {
        FinalClassA fa1, fa2;
        fa1 = new FinalClassA();
        try {
            fa2= new FinalClassA(5);
        }
        catch (FinalClassAException fcae) {
            System.out.println(fcae.getMessage());
        }
        catch (IllegalArgumentException iae) {
            System.out.println(iae.getMessage());
        }
        finally {
            System.out.print("Let's keep going...");
        }
    }
}

public class FinalMainB {
    static FinalClassA[] myClassA= new FinalClassA[4];
    static int[] nums = {2,5,-3,7};
    public static void main (String[] args) {

        sub1();
        System.out.println("*****");
        sub2();
        System.out.println("*****");
        sub3();
    }
    public static void sub1() {
        for (int i=0; i<myClassA.length; i++)
            try {
                if (nums[i]%2==0)
                    myClassA[i] = new FinalClassA(nums[i]);
                else
                    myClassA[i] = new FinalClassB(nums[i]);
            }
            catch (FinalClassAException e2cae) {
                myClassA[i] = new FinalClassA();
            }
    }
    public static void sub2() {
        FinalClassB f2cb = new FinalClassB();
        for (int i=0; i<myClassA.length-1; i++) {
            System.out.println(myClassA[i].sum(nums[i],nums[i+1]));
        }
    }
    public static void sub3() {
        FinalClassB f2cb = new FinalClassB();
        System.out.println(f2cb.sum(10,20,f2cb.times(4,5)));
    }
}

public class FinalMainC {

    public static void main(String[] args) {
        FinalClassB e2b1, e2b2;
        e2b1 = new FinalClassB();
        e2b2 = new FinalClassB(e2b1.sum(2,3,4));
    }
}
```