

**CS 212 – Object-Oriented Programming in Java**  
**Practice Final Exam A Solutions**

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

**Directions:** There are six 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. (12 points)**

```
public class Q1 {
    private static int ourCount = 0;
    private int count;
    public Q1 (int n) {
        ourCount++;
        count=n;
    }
    public String toString() {
        return "Q1: "+count+" of "+ourCount;
    }
}

public class Q1Main {
    private static final int numberOfQs = 3;
    public static void main(String args[]) {
        Q1 QArray[] = new Q1[numberOfQs];
        for (int i=0; i<QArray.length; i++){
            QArray[i] = new Q1(i+1);
            System.out.println(QArray[i]);
        }
        System.out.println("*****");
        for (int i=0; i<QArray.length; i++){
            System.out.println(QArray[i]);
        }
    }
}
```

What is the output of class Q1Main:

```
Q1: 1 of 1
Q1: 2 of 2
Q1: 3 of 3
*****
Q1: 1 of 3
Q1: 2 of 3
Q1: 3 of 3
```

**Question 2. (12 points)**

```
public class Q2 extends Q1 {
    private int myNumber;
    public Q2(){
        super(0);
        myNumber=0;
    }
    public Q2(int n, int m){
        super(n);
        myNumber=n+m;
    }
    public String toString(){
        return super.toString()+" "+
            getClass().getName()+" "+myNumber;
    }
}

public class Q2Main {
    private static final int numberOfQs = 4;
    public static void main(String args[]) {
        Q2 QArray[] = new Q2[numberOfQs];
        for (int i=0; i<QArray.length; i++){
            if(i%2==0)
                QArray[i] = new Q2(i+1,i+numberOfQs);
            else
                QArray[i] = new Q2();
        }
        for (int i=0; i<QArray.length; i++){
            System.out.println(QArray[i]);
        }
    }
}
```

What is the output of class Q2Main:

Q1: 1 of 4 Q2: 5  
Q1: 0 of 4 Q2: 0  
Q1: 3 of 4 Q2: 9  
Q1: 0 of 4 Q2: 0

**Question 3. 12 points**

```
public class Q3Main {
    private static final int numberOfQs = 4;
    public static void main(String args[]) {
        Q1 QArray[] = new Q1[numberOfQs];
        for (int i=0; i<QArray.length; i++){
            try {
                if(i%2==0)
                    QArray[i] = new Q2(i-2,i+numberOfQs);
                else
                    QArray[i] = new Q2();
            }
            catch (Q1Exception q1e){
                System.out.println(q1e.getMessage());
                QArray[i]= new Q1(99);
            }
        }
        for (int i=0; i<QArray.length; i++){
            System.out.println(QArray[i]);
        }
    }
}
```

What will be the output of class Q3Main:

n cannot be negative.  
Q1: 99 of 4  
Q1: 0 of 4 Q2: 0  
Q1: 0 of 4 Q2: 6  
Q1: 0 of 4 Q2: 0

**Question 4. 12 points**

Suppose the following Interface is defined:

```
public interface QInterface {  
    public String toString();  
    public int getN();  
}
```

And class Q2 now implements that interface:

```
public class Q2 extends Q1 implements QInterface {  
    private int myNumber;  
    public Q2() {  
        super(0);  
        myNumber=0;  
    }  
    public Q2(int n, int m){  
        super(n);  
        myNumber=n+m;  
    }  
    public String toString(){  
        return super.toString()+" "+  
            getClass().getName()+" "+myNumber;  
    }  
}
```

What changes need to be made to Q2?

Q2 must provide the method `int getN();`  
Q2 does not have to provide `String toString();`

What problem, if any, would be caused by putting the following method into class Q2?

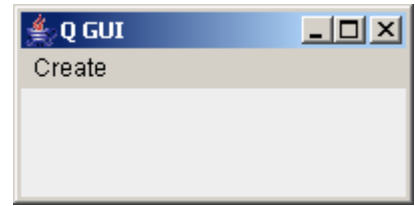
```
public int getCount() {  
    return count;  
}
```

"count" is declared private in Q1, so Q2 does not have access to it.

**Question 5. (24 points)**

The code on the next page is intended to create a GUI (Such as the one at the right) with one menu called "Create", and three menu choices, "New ArraySequence", "New SkippedArraySequence", and "Quit". All three menu choices are handled by a single `ArraySequenceListener` object.

Write appropriate code to fill in the boxes on each of the following lines:



- 3. `JFrame` \_\_\_\_\_
- 6. `JMenu` \_\_\_\_\_
- 11. `JMenuItem` \_\_\_\_\_
- 16. `add(mQ2)` \_\_\_\_\_
- 19. `JMenuBar` \_\_\_\_\_
- 22. `QListener` \_\_\_\_\_
- 24. `createMenuML` \_\_\_\_\_
- 34. `ActionListener` \_\_\_\_\_
- 37. `f` \_\_\_\_\_
- 39. `e` \_\_\_\_\_
- 41. `"New Q1"` \_\_\_\_\_
- 46. `Q2` \_\_\_\_\_

```
1  import java.awt.*;
2  import javax.swing.*;
3  public class QGUI extends [ ] {
4      private JFrame QGUI;
5      private JMenuBar mainMenuBar;
6      private [ ] createMenu;
7      private JMenuItem mQ1, mQ2, mQuit;
8      private QListener createMenuML;
9      public QGUI (String title) {
10         QGUI = new JFrame(title);
11         mQ1 = new [ ] ("New Q1");
12         mQ2 = new JMenuItem ("New Q2");
13         mQuit = new JMenuItem ("Quit");
14         createMenu = new JMenu ("Create", /* tearoff = */ false);
15         createMenu.add(mQ1);
16         createMenu.addSeparator();
17         createMenu.add(mQuit);
18         mainMenuBar = new [ ];
19         mainMenuBar.add(createMenu);
20         QGUI.setMenuBar(mainMenuBar);
21         createMenuML = new [ ](this);
22         mQ1.addActionListener(createMenuML);
23         mQ2.addActionListener([ ]);
24         mQuit.addActionListener(createMenuML);
25         QGUI.setDefaultCloseOperation(EXIT_ON_CLOSE);
26         QGUI.setSize(200,100);
27         QGUI.setLocation(100,100);
28         QGUI.setVisible(true);
29     } // constructor
30 } // class

32 import java.awt.*;
33 import java.awt.event.*;
34 public class QListener implements [ ] {
35     private QGUI mainFrame;
36     public QListener(QGUI f) {
37         mainFrame = [ ];
38     } // constructor
39     public void actionPerformed (ActionEvent [ ]) {
40         String chosenItem = ((JMenuItem) e.getSource()).getLabel();
41         if (chosenItem.equals([ ])) {
42             Q1 q1 = new Q1(5);
43             System.out.println("New Q1 is created...");
44         } // if
45         else if (chosenItem.equals("New Q2")) {
46             Q2 q2 = new [ ](2,8);
47         } // if
48         else if (chosenItem.equals("Quit")) {
49             System.exit(0);
50         }
51     } // method actionPerformed(ActionEvent)
52 } // class QListener
```

**Question 6. (20 points)**

The following code represents a doubly-linked list node, and a doubly-linked list with head node. Fill in the missing code in the boxes below.

```
public class Q1Node {  
    public Q1 data;  
    public Q1Node next;  
    public Q1Node previous;  
    public Q1Node (Q1 data) {
```

```
        this.data = data;  
        previous = null;  
        next = null;
```

```
    }
```

```
}  
public class Q1List {  
    private Q1Node first,last;  
    private int numberOfNodes;  
    public Q1List() {
```

```
        Q1Node temp = new Q1Node(null);  
        first = temp;  
        last = temp;  
        numberOfNodes=0;
```

```
    }
```

```
    public append(Q1 qdata){
```

```
        Q1Node temp = new Q1Node(qdata);  
        last.next = temp;  
        last = temp;  
        numberOfNodes++;
```

```
    }
```

```
    public void printList() {
```

```
        Q1Node temp = first.next;  
        while (temp != null) { //or temp!=last  
            System.out.println(temp.data);  
            temp = temp.next;  
        }
```

```
    }
```

```
}
```

**Question 7. 8 points**

Rewrite the method `printList` from question 6 recursively.

Assume its parameter is reference to the first real node on the list (not the head node).

```
public printList(QNode p)
```

```
if (p != null) {  
    System.out.println(p.data);  
    printList (p.next);  
}
```