Practice Exercises: Teaching Block 3

- Interfaces
- GUI
- Garbage collection
- Strings
- Other practice exercise types: “Fill in the gaps”

This set of exercises is in addition to those included directly in lecture slides (and extra reading materials), which you should also attempt.

Question 1

- Consider an interface named **Colorable**, as follows:

```
public interface Colorable { public void howToColor(); }
```

  - Create a class named **Square** that extends **GeometricObject** *and* implements **Colorable**.
  - Implement **howToColor()** to display a message on how to colour the square.
  - Create an additional class to test the creation of a **Square** and its method **howToColor()**.

```
public abstract class GeometricObject {
  // some methods and instance variables

  public abstract double findArea();
  public abstract double findPerimeter();
}
```

Question 2

- Answer the following questions:
  - Identify what is wrong with the **`interface`** below.

```
public interface SomethingIsWrong {
  void aMethod(int aValue) {
    System.out.println("Hi Mom");
  }
}
```

  - Can an interface be given the **`private`** access modifier?

Question 3

- Identify the statements below about interfaces, that are TRUE.
  a. Interfaces do not allow for multiple inheritance at design level.
  b. Interfaces can be extended by any number of other interfaces.
  c. Interfaces can extend any number of other interfaces.
  d. Members of an interface are never `static`.
  e. Methods of an interface can always be declared `static`.

  Explain your answer.

Star Q1

Question 4

- The following program is supposed to display a button in a frame (as shown), but nothing is displayed. What is the problem?

```
import javax.swing.*;

public class Test extends JFrame {
  public Test() {
    getContentPane().add(new JButton("OK"));
  }

  public static void main(String[] args) {
    JFrame frame = new JFrame();
    frame.setSize(100,200);
    frame.setVisible(true);
  }
}
```

Question 5

- What happens when the code below is run? Will anything be displayed?

```
import java.awt.*;
import javax.swing.*;
public class Test extends JFrame {
  public Test() {
  JButton jbt1 = new JButton();
  JButton jbt2 = new JButton();
  JPanel p1 = new JPanel();
  p1.add(jbt1);
  JPanel p2 = new JPanel();
  p2.add(jbt2);
  JPanel p3 = new JPanel();
  p2.add(jbt1);
  getContentPane().add(p1, BorderLayout.NORTH);
  getContentPane().add(p2, BorderLayout.SOUTH);
  getContentPane().add(p3, BorderLayout.CENTER);
}
```

(code cont.)

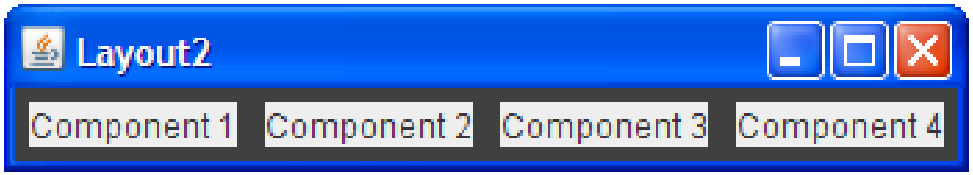
```
  public static void main(String[] args) {
    // Create a frame and set its properties.
    JFrame frame = new Test();
    frame.setTitle("ButtonIcons");
    frame.setSize(220,120);
    frame.setDefaultCloseOperation(
                    JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
  }
}
```

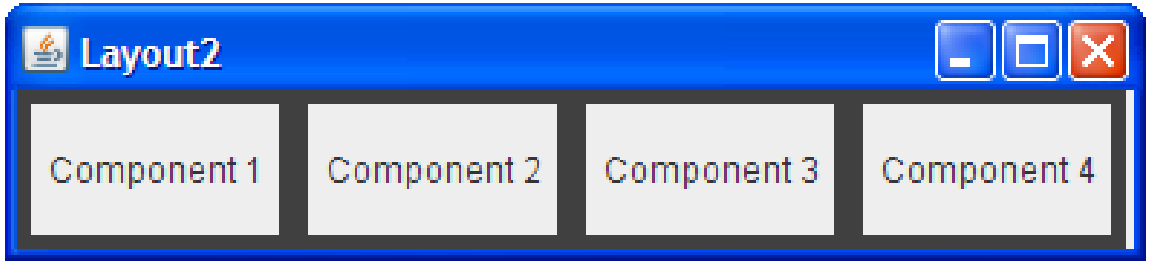
Question 6

- Choose the layout manager(s) most naturally suited for the following layout description, an example of which is given below: “the container has a row of components that should all be displayed at the same size, filling the container’s entire area”. **Explain** your choice.

  a. **`FlowLayout`**
  b. **`GridLayout`**
  c. **`BorderLayout`**
  d. Options *a* and *b*.

Note: You can assume that the container controlled by the layout manager is a **`JPanel`**.





Star Q2

Question 7

- Identify the lines of code that, if added to the program at point **X** would cause exactly one more object to be eligible for the **Garbage Collector**.

```
copyGC = null;
gc2 = null;
newGC = gc3;
gc1 = null;
newGC = null;
gc4 = null;
gc3 = gc2;
gc1 = gc4;
gc3 = null;
```

```
public class GC {
  public static GC doStuff() {
    GC newGC = new GC();
    doStuff2(newGC);
    return newGC;
  }
  public static void main(String[] args) {
     GC gc1;
     GC gc2 = new GC();
     GC gc3 = new GC();
     GC gc4 = gc3;
     gc1 = doStuff();
     X
     // call more methods
  }
  public static void doStuff2(GC copyGC) {
    GC localGC;
  }
}
```

Note that variable **gc1** is not initialised to a default value here, because it is a local variable.

Question 8

- Identify the location where the object, initially referenced with **`arg1`**, is eligible for garbage collection. **Explain**.

```
public class MyClass {
  public static void main(String[] args) {
    String msg;
    String pre = "This program was called with ";
    String post = " as first argument.";
    String arg1 = new String((args.length > 0) ? "'" +
                  args[0] + "'" : "<no argument>");
    msg = arg1; arg1 = null; // (1)
    msg = pre + msg + post; // (2)
    pre = null; // (3)
    System.out.println(msg); msg = null; // (4)
    post = null; // (5)
    args = null; // (6)
  }
}
```

Question 9

- Determine the order in which the constructors execute in this example.

```java
//Should be in C1.java
public class C1 {
    public C1() {
        System.out.println("1");
    }
}
//Should be in C2.java
public class C2 extends C1{
    public C2() {
        super();
        System.out.println("2");
    }
}
//Should be in C3.java
public class C3 extends C2 {
    public C3() {
        System.out.println("3");
    }

    public static void main(String args[]) {
        //Q: What list of numbers will be printed?
        //   (What order are the constructors executed?)
        C3 obj = new C3();
    }
}
```

Question 10

- Determine what is wrong or missing and what is OK, in the code below.

```java
public class L2Super {
    private String name;
    private int num = 0;
    public void setName(String aName) {
        name = aName;
    }
    public void setNum(int num) {
        num = num;
    }
    public L2Super(String aName) {
        name = aName;
    }
    public int getNum() { return num; }
    public String getName() { return name; }
}
```

```java
public class L2Sub extends L2Super {

    public L2Sub() {    }
    public L2Sub(String aName){
        super(aName);
    }

    public String getName() { return name; }

    public static void main(String args[]){
        L2Sub a = new L2Sub();
        L2Sub b = new L2Sub("Tim");
        System.out.println(b.name);
        System.out.println(b.getName());

        b.setNum(5);
        System.out.println(b.getNum());
    }
```

Question 11

- Consider the following string:

```
String hannah = "Did Hannah see bees? Hannah did.";
```

  - What is the value displayed by the expression `hannah.length()`?
  - What is the value returned by the method call `hannah.charAt(12)`?
  - Write an expression that refers to the letter `b` in the `String` referred to by `hannah`.

Questions 12

- Write a program that *computes your initials* from your full name and *displays them*.

Question 13

- In the program below, what is the value of **`result`** after each *numbered line* executes?

```
  public class ComputeResult {
    public static void main(String[] args) {
      String original = "software";
      StringBuilder result = new StringBuilder("hi");
      int index = original.indexOf('a');
/*1*/ result.setCharAt(0, original.charAt(0));
/*2*/ result.setCharAt(1, original.charAt(original.length()-1));
/*3*/ result.insert(1, original.charAt(4));
/*4*/ result.append(original.substring(1,4));
/*5*/ result.insert(3, (original.substring(index, index+2) + " "));
      System.out.println(result);
    }
  }
```

Question 14

Star Q3

- Which two statements are TRUE?
  a. `String` objects are immutable.
  b. Subclasses of the `String` class can be mutable.
  c. All wrapper classes are declared `final`.
  d. All objects have a `private` method named `toString()`.

Exercise: Fill in the Gaps

Gaps **B–E** are single ‘words’/values.

```
public ___(1)___ ShortConcept {
  int a = 20;
  int b = 30;
  void concept1();
}
```

```
public class ShortExample
        ___(2)___ ShortConcept {
  public void concept1() {
    a = 1;
    int sum = a + b;
    System.out.println("Sum is " + sum);
  }
}
```

If `(1)` is `class`, then compiling `ShortConcept` will generate a compiler error because _____**A**_____. This error can be prevented by adding the keyword ____**B**____ in front of `class` and in front of `void concept1();`. In this case, `(2)` should be the keyword ____**C**____. `ShortExample` would then compile and the method `concept1()` would produce a value of ____**D**____.

If `(1)` is `interface`, then compiling `ShortConcept` will not generate any errors. In this case, `(2)` should be the keyword ____**E**____. However, `ShortExample` would generate a compiler error because ____**F**____.