

# Written Exam

## Programming Mobile Applications Medialogy, 2nd semester

Tuesday 13 June 2023, 9.00 – 13.00

Navn: \_\_\_\_\_

Studienr: \_\_\_\_\_

## Programming Mobile Applications

### Examination

13 June 2023, 9.00 – 13.00

### Instructions

- You have 4 hours to complete this examination.
- Neither electronic devices nor written material are allowed in the examination room.
- This examination consists of 10 questions. Each question is worth 10 marks. You must obtain at least 50 marks to pass.
- Do not write any answers on this question paper—answers written on the question paper will be ignored by the examiner. Write all your answers on the writing paper provided.
- Do not write your answers in pencil and do not use a pen with red or green ink. Use a pen with blue or black ink.
- Hand in no more than one answer to each question.
- Do not turn over until you are told to do so by the invigilator.

## Question 1

For each of the following statements, write down whether it is true or false.

- a) A module encapsulates an interface.
- b) If a module A is a client of a module B, then a change in the public interface of B may necessitate a change in module A.
- c) A module can have at most one interface.
- d) If a module A provides services to a module B, then this implies that B depends on A.
- e) A module exhibits high cohesion if it is easy to decompose it into smaller modules.
- f) In software engineering, an object is a thing that has behaviour, state and identity.
- g) Two methods within the same class can have the same selector.
- h) A class's public interface contains its private interface.
- i) If a class, A, is a subclass of another class, B, then any method defined in the public interface of B can also be called on an object of class A.
- j) If p is a variable that refers to an object of class x, then p may refer to any object from any subclass of x.

[1 mark for each correct part]

## Question 2

Suppose we have the following Java program defined in a file called Question2.java:

```
1 package com.titanmusic.pmaexam2023;
2
3 public class Question2 {
4     public static void main(String[] args) {
5         if (args.length > 1)
6             for(int i = args.length-1; i >= 0; i--)
7                 if (args[i].length() < 3)
8                     System.out.print(args[i]+" ");
9     }
10 }
```

Now suppose we export this Java class to a runnable jar file called Question2.jar and that we then run this runnable jar file by writing the following command at the command prompt:

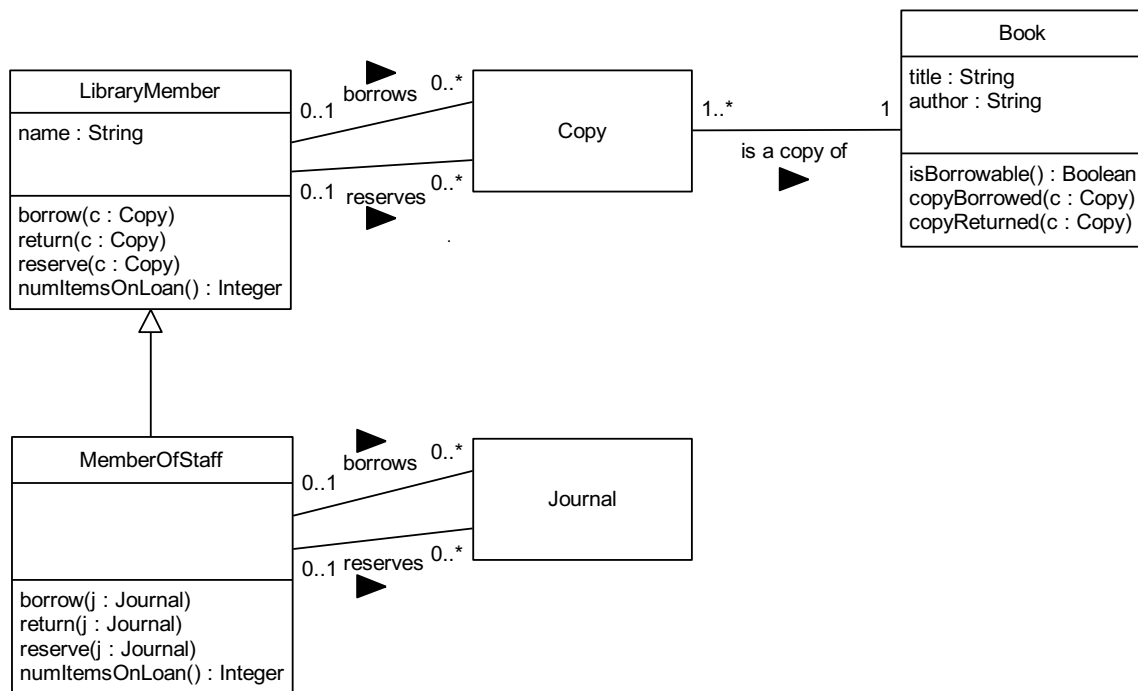
```
% java -jar Question2.jar aaa bb ccc dd eee
```

What will the program output to the console window?

[10 marks]

### Question 3

Study the following UML class diagram and answer the questions that follow it.



- Give an example from the diagram of *method overloading*.
- Give an example from the diagram of *method overriding*.
- With how many **Book** objects can each **Copy** object be associated?
- What is the maximum number of **MemberOfStaff** objects that can be associated with each **Journal** object?
- Can **LibraryMember** objects be associated with **Journal** objects?

[2 marks for each correct part]

## Question 4

What does the following Java program print to the console when it is executed?

```
1 package com.titanmusic.pmaexam2023;
2
3 class One {
4     private int x, y;
5     private String s;
6     public One() {x = y = 0; s = "";}
7     public One(int x, int y, String s) {setX(x); setY(y); setS(s);}
8     public void setX(int x) {this.x = x;}
9     public void setY(int y) {this.y = y;}
10    public void setS(String s) {this.s = s;}
11    public int getX() {return x;}
12    public int getY() {return y;}
13    public String getS() {return s;}
14    public String toString() {return getX()+" "+getY()+" "+getS();}
15    public void doubleVals() {
16        setX(2*getX()); setY(2*getY()); setS(getS()+getS());
17    }
18 }
19
20 class Two extends One {
21     private int z;
22     private String t;
23     public Two() {super(); z = 0; t = "";}
24     public Two(int x, int y, int z, String s, String t) {super(x,y,s); setZ(z); setT(t);}
25     public void setZ(int z) {this.z = z;}
26     public void setT(String t) {this.t = t;}
27     public int getZ() {return z;}
28     public String getT() {return t;}
29     public String toString() {return super.toString()+" "+getZ()+" "+getT();}
30     public void doubleVals() {super.doubleVals(); setZ(2*getZ()); setT(getT()+getT());}
31 }
32
33 public class Question4 {
34     public static void main(String[] args) {
35         One one = new One(1,2,"Yes");
36         Two two = new Two(3,4,5,"No","Maybe");
37         one.doubleVals();
38         two.doubleVals();
39         System.out.println(one);
40         System.out.println(two);
41     }
42 }
```

[10 marks]

## Question 5

Study the following Java code, which shows the contents of a file called Question5.java, then answer the questions that follow it.

```
1 package com.titanmusic.pmaexam2023;
2
3 import java.util.TreeSet;
4
5 class Point {
6     public double x, y;
7     public Point(double x, double y) {this.x = x; this.y = y;}
8     public void move(double x, double y) {this.x += x; this.y += y;}
9     public String toString() {return "("+x+","+y+"");}
10 }
11
12 abstract class Shape implements Comparable<Shape> {
13     public Point refPoint;
14     public abstract double getArea();
15     public void move(double x, double y) {refPoint.move(x, y);}
16     public String toString() {return refPoint+", "+getArea();}
17     public int compareTo(Shape shape) {
18         if (shape == null) return 1;
19         double d = refPoint.x-shape.refPoint.x;
20         if (d != 0.0) return (int)(Math.signum(d));
21         d = refPoint.y-shape.refPoint.y;
22         if (d != 0.0) return (int)(Math.signum(d));
23         d = getArea()-shape.getArea();
24         return (int)(Math.signum(d));
25     }
26     public boolean equals(Object obj) {
27         if (obj==null) return false;
28         if (!(obj instanceof Shape)) return false;
29         return compareTo((Shape)obj)==0;
30     }
31 }
32
33 class Circle extends Shape {
34     private double radius;
35     public Circle(double x, double y, double radius) {refPoint = new Point(x,y); this.radius = radius;}
36     public double getArea() {return 3 * radius * radius;}
37     public String toString() {return "Circle: "+super.toString();}
38 }
39
40 class Rectangle extends Shape {
41     private Point topLeft, bottomRight;
42     public Rectangle(double tlx, double tly, double brx, double bry) {
43         refPoint = new Point(tlx,tly);
44         bottomRight = new Point(brx,bry);
45     }
46     public double getArea() {return (topLeft.x-bottomRight.x)*(bottomRight.y-topLeft.y);}
47     public String toString() {return "Rectangle: "+super.toString();}
48 }
49
50 public class Question5 {
51     public static void main(String[] args ) {
52         TreeSet<Shape> shapes = new TreeSet<Shape>();
53         shapes.add(new Rectangle(1,3,2,0));
54         shapes.add(new Rectangle(2,1,3,0));
55         shapes.add(new Circle(1,3,1));
56         shapes.add(new Circle(1,3,2));
57         for(Shape s : shapes)
58             System.out.println(s);
59     }
60 }
```

- Give an example from the code above of a method definition that overrides a method defined in a superclass. Give the name of the superclass within which the overridden method is defined and the line number where the overriding method is defined.
- Give an example from the code above of a method that calls the overridden method defined in the superclass. State the line number where the call occurs.
- Give an example from Question5.java of a class definition that implements an interface. Name both the class and the interface that it implements.
- Give an example from the code of a class that extends an abstract superclass. Name both the class and its abstract superclass.
- Give an example of a single abstract method being implemented in two different subclasses defined in Question5.java. Give the numbers of the lines in which the abstract method is declared and implemented.

[2 marks for each correct part]

## Question 6

This question also refers to the Question5.java file shown above in Question 5.

- Give an example of a polymorphic variable in the code in Question5.java and state the line in which it is first declared.
- Write down the area, as computed by the appropriate getArea() method, of each of the four Shape objects constructed in lines 53-56 of the Question5.java file shown in Question 5.
- Write down the String that will be output by the relevant toString() method for each of the four Shape objects constructed in lines 53-56 of the Question5.java file shown in Question 5.
- Are any of the Shape objects constructed in lines 53-56 "equal", where we define Shape equality by the equals() method in lines 26-30 of Question5.java?
- What does the Question5.java program print to the console when it is executed?

[2 marks for each correct part]

## Question 7

The following program does not necessarily print out the same thing every time it is run. Write down two different possible outputs that the following program might print out.

[10 marks]

```
public class Question7 {
    static boolean GREEN_LIGHT = false;
    static boolean FINISHED = false;
    static int TOTAL = 5;
    static int DELAY = 500;
    static Random R = new Random();
    public static void main(String[] args) {
        Thread green = new Thread(new Runnable() {
            public void run() {
                while(!FINISHED) {
                    if (GREEN_LIGHT) {
                        System.out.print(--TOTAL+" ");
                        if (TOTAL == 0) {
                            FINISHED = true;
                            System.out.println("Green thread won!");
                            return;
                        }
                    }
                    try {
                        Thread.sleep(R.nextInt(DELAY));
                        GREEN_LIGHT = true;
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        Thread red = new Thread(new Runnable() {
            public void run() {
                while(!FINISHED) {
                    if (!GREEN_LIGHT) {
                        System.out.print(++TOTAL+" ");
                        if (TOTAL == 10) {
                            FINISHED = true;
                            System.out.println("Red thread won!");
                            return;
                        }
                    }
                    try {
                        Thread.sleep(R.nextInt(DELAY));
                        GREEN_LIGHT = false;
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
        green.start();
        red.start();
    }
}
```

## Question 8

Study the following Java class definition and answer the questions that follow it.

```
--
12 public class Question8 extends JFrame implements ActionListener {
13
14     private static final long serialVersionUID = 1L;
15     JButton button = new JButton("Press me");
16     JLabel label = new JLabel("Bing");
17
18     public Question8() {
19         setDefaultCloseOperation(EXIT_ON_CLOSE);
20         getContentPane().add(button);
21         getContentPane().add(label, BorderLayout.LINE_START);
22         button.addActionListener(this);
23         pack();
24         setVisible(true);
25     }
26
27     public void actionPerformed(ActionEvent e) {
28         if (label.getText().equals("Bing"))
29             label.setText("Bang");
30         else
31             label.setText("Bing");
32     }
33
34     public static void main(String[] args) {
35         SwingUtilities.invokeLater(new Runnable() {
36             public void run() {
37                 new Question8();
38             }
39         });
40     }
41 }
```

- In line 37, why is the Question8 object constructed inside a Runnable object and given as an argument to the *invokeLater* method?
- What is the purpose of the BorderLayout.LINE\_START argument in line 21? What would happen if it were removed?
- Sketch what the GUI will look like when the program is run and describe what happens when the button is pressed.
- What is the purpose of the call to the pack() method in line 23?
- How would the GUI behave if line 22 were deleted?

[2 marks for each correct part]



## Question 9

Study the following Android class definition and answer the questions that follow it.

```
public class SensorActivity extends Activity implements SensorEventListener {
    private SensorManager mSensorManager;
    private Sensor mLight;

    @Override
    public final void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mLight = mSensorManager.getDefaultSensor(Sensor.TYPE_LIGHT);
    }

    @Override
    public final void onAccuracyChanged(Sensor sensor, int accuracy) {
        // Do something here if sensor accuracy changes.
    }

    @Override
    public final void onSensorChanged(SensorEvent event) {
        // The light sensor returns a single value.
        // Many sensors return 3 values, one for each axis.
        float lux = event.values[0];
        // Do something with this sensor value.
    }

    @Override
    protected void onResume() {
        super.onResume();
        mSensorManager.registerListener(this, mLight, SensorManager.SENSOR_DELAY_NORMAL);
    }

    @Override
    protected void onPause() {
        super.onPause();
        mSensorManager.unregisterListener(this);
    }
}
```

- Which lifecycle callback methods are overridden in this class definition? [2 marks]
- Does this activity use a streaming sensor or a non-streaming sensor? How do you know? [2 marks]
- What is the purpose of the line, `mSensorManager.unregisterListener(this)` in the `onPause` method and why is this line present in this particular method? [3 marks]
- Which object listens for events emitted by the sensor? What does this object do in response to a sensor event being emitted? [3 marks]

## Question 10

Study the following code which defines an Android Activity, and answer the questions that follow it.

```
12 public class MainActivity extends AppCompatActivity {
13
14     static int requestCode = 1234567;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20     }
21
22     public void respondToButton(View view) {
23         Toast.makeText(context: this, text: "Button was pressed", Toast.LENGTH_LONG).show();
24         Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
25         if (intent.resolveActivity(getPackageManager()) != null) {
26             startActivityForResult(intent, requestCode);
27         }
28     }
29
30     public void onActivityResult(int requestCode, int resultCode, Intent data) {
31         if (requestCode == MainActivity.requestCode && resultCode == RESULT_OK) {
32             Bundle extras = data.getExtras();
33             Bitmap imageBitmap = (Bitmap) extras.get("data");
34             ImageView imageView = (ImageView) findViewById(R.id.imageView);
35             imageView.setImageBitmap(imageBitmap);
36         }
37     }
38 }
```

- This code uses an intent to start another activity. Write down the line number of the line in which this intent is instantiated and state whether the intent is implicit or explicit. [1 mark]
- Which of the three methods defined in the example is a *lifecycle callback*? [1 mark]
- What is the purpose of the `requestCode` variable defined in line 14? [2 marks]
- The following is an excerpt from the XML layout file that defines the layout associated with the `MainActivity` class defined above:

```
<Button
    android:onClick="respondToButton"
    android:id="@+id/button"
```

- When the button referred to in this excerpt is pressed, the `respondToButton` method is called. When this happens, what object is passed to the `respondToButton` method as its argument? [1 mark]
  - What is the purpose of the button's "android:id" attribute? Explain the meaning of the "@", "+" and "id/" components of the value of this attribute. [2 marks]
- An intent contains a `Bundle` object that is accessed through its `getExtras()` method. Explain what a `Bundle` is and how it is used in the Java code given above. [2 marks]
  - In line 33, why is the right-hand side of the assignment typecast to a `Bitmap`? [1 mark]

END OF EXAMINATION