

CONFIDENTIAL



**UTHM**  
Universiti Tun Hussein Onn Malaysia

**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**FINAL EXAMINATION  
SEMESTER II  
SESSION 2022/2023**

COURSE NAME : JAVA PROGRAMMING  
COURSE CODE : BIT 33803  
PROGRAMME CODE : BIT  
EXAMINATION DATE : JULY/AUGUST 2023  
DURATION : 3 HOURS  
INSTRUCTION :  
1. ANSWER ALL QUESTIONS  
2. THIS FINAL EXAMINATION IS CONDUCTED VIA **CLOSED BOOK.**  
3. STUDENTS ARE **PROHIBITED** TO CONSULT THEIR OWN MATERIAL OR ANY EXTERNAL RESOURCES DURING THE EXAMINATION CONDUCTED VIA CLOSED BOOK.

THIS QUESTION PAPER CONSISTS OF **FIVE (5)** PAGES

CONFIDENTIAL

**TERBUKA**

**Q1** Indicate whether each of the following statements is **TRUE** or **FALSE**.

- (a) Classes from which objects can be instantiated are called concrete classes. (2 marks)
- (b) All methods in an abstract class must be declared as abstract methods. (2 marks)
- (c) In Java, invoking a subclass-only method through a superclass instance is allowed. (2 marks)
- (d) If a superclass declared an abstract method, a subclass that extends it must implement that method. (2 marks)
- (e) An object of a class that implements an interface is known as an object of that interface type. (2 marks)

**Q2** Determine whether each of the following contains **Error** or **No Error**. Justify your answers.

- (a) `buttonName = JButton("Caption");` (2 marks)
- (b) 

```
abstract public void getName(){
    return name;
}
```

 (2 marks)
- (c) `txtField = new JTextField (50, "Default Text");` (2 marks)
- (d) 

```
//creating JFrame and display
JFrame f = new JFrame("A window");
f.setVisible(true);
```

 (2 marks)
- (e) 

```
public interface AnimalSound {
    public void Cat() {
        System.out.println("Meow.. Meow..");
    }
}
```

 (2 marks)

**TERBUKA**

*[Faint, illegible text in the bottom right corner]*

**Q3** Determine the output of the following segment code:

```
(a) public static void main(String[] args) {
        int sum = 0;
        int i=3;
        while (i<10) {
            sum = sum +(i*i);
            i++;
            i++;
        }
        System.out.println("sum is : "+sum);
    }
```

(5 marks)

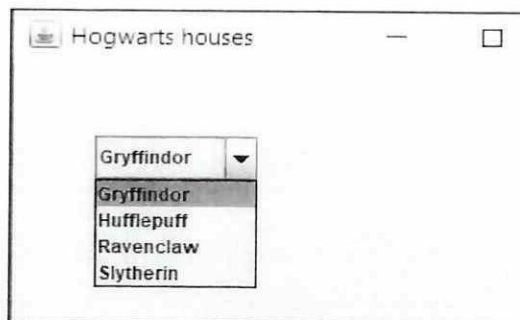
```
(b) public static void main(String[] args) {
        for (int i=-2; i<=2;i++) {
            for(int j=-2; j<=2; j++) {
                if(i<j)
                    System.out.print("7");
                else
                    System.out.print("3");
            }
            System.out.println();
        }
    }
```

(5 marks)

**Q4** Write a segment code in Java for the following:

(a) Create the user interface component as shown as in **Figure Q4(a)**.

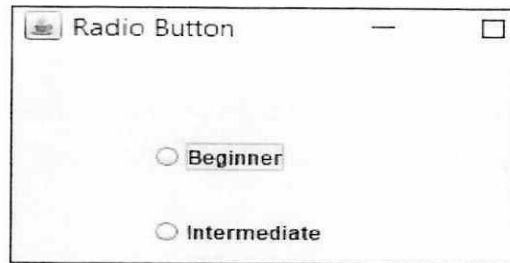
(12 marks)



**Figure Q4(a)**

- (b) Create a group of radio buttons with the labels “Beginner” and “Intermediate” as shown as in **Figure Q4(b)**.

(8 marks)



**Figure Q4(b)**

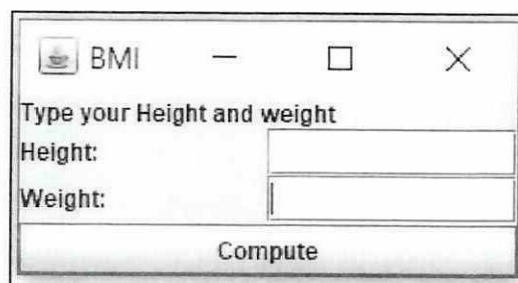
- Q5** Apply exception handling for the segment code in **Figure Q5** by using `try.. catch` block and `showMessageDialog()`.

(10 marks)

```
public class Age {
    public static void main(String[] args) {
        int age;
        age = Integer.parseInt(JOptionPane.showInputDialog("How
            old are you?"));
        System.out.println("Your age is:" +age);
    }
}
```

**Figure Q5**

- Q6** Answer **Q6(a) – Q6(c)** based on the GUI in **Figure Q6(a)** and its incomplete program codes in **Figure Q6(b)**.



**Figure Q6(a)**

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class BMI implements ActionListener{
//Write Swing GUI component declaration

private final JFrame frame;

    public BMI() {

        //Write code statements to set up GUI components

        computeButton.addActionListener(this);
        frame = new JFrame("BMI");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLayout(new BorderLayout());
        frame.add(bmiLabel, BorderLayout.NORTH);
        frame.add(center, BorderLayout.CENTER);
        frame.add(computeButton, BorderLayout.SOUTH);
        frame.pack();
        frame.setVisible(true);
    }

    public void actionPerformed(ActionEvent event) {

        //write code statements to
        //Handle clicks on compute button to calculate the BMI
        //Read height and weight data from text fields

    }

    public static void main (String args[]) {
        BMI gui = new BMI();
    }
}

```

Figure Q6(b)

- (a) Identify the GUI components in **Figure Q6(a)** and write the swing GUI components declaration statements for the program. (4 marks)
- (b) Write the appropriate code statements to set up the GUI components as depicted in **Figure Q6(a)**. (10 marks)
- (c) Write the method implementation for `actionPerformed()`. The formula to calculate Body Mass Index (BMI) is given as follows:

$$BMI = \frac{weight}{height^2} \times 703$$

(6 marks)

-END OF QUESTIONS -