Open Book Assignment 2 **Name: Venkatesh Dhongadi**

OOPS with JAVA (18CS34) **USN: 2GI19CS175**

1)

Unit 2(Methods and classes)

1. **Fill in the blanks**
    - A__________ is an instance of a class.
    - Finalize() is the method of ______ class
2. **True/False**
    - You can gain access to the hidden instance variable by referring to it through this. (T/F)
    - Protected are also called inheritance level access modifiers (T/F)
3. **Multiple choice questions**
    - A Java method is comparable to a __ in c language.

      A) structure

      B) union

      C) function

      D) Enum

    - 6) In Java, a variable name cannot start with a___.

      A) number

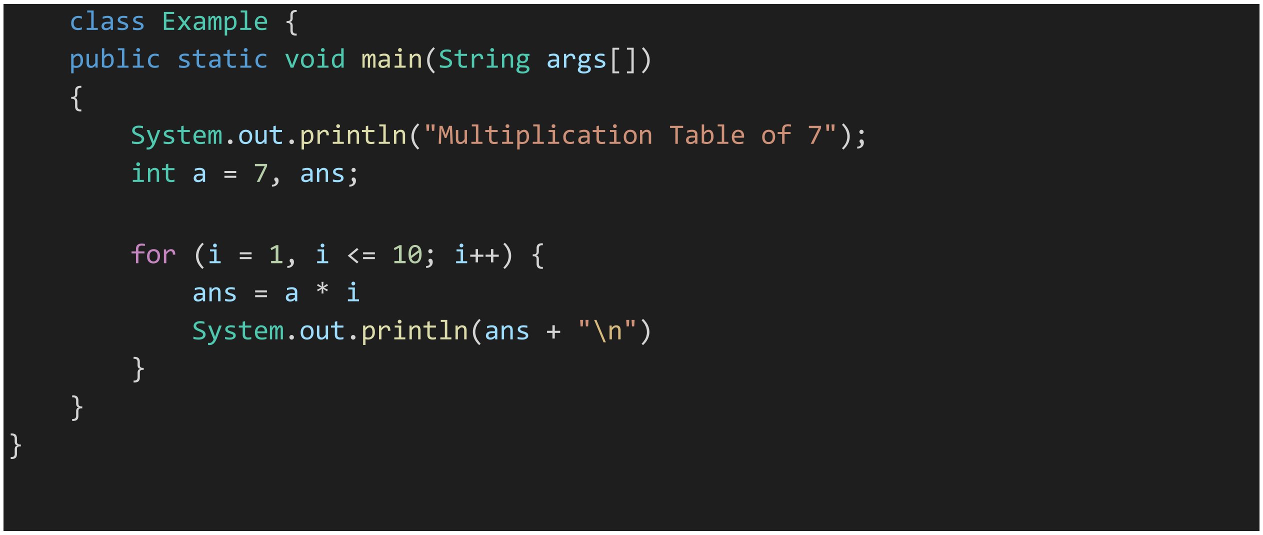
      B) # (pound)

      C) - (hyphen)

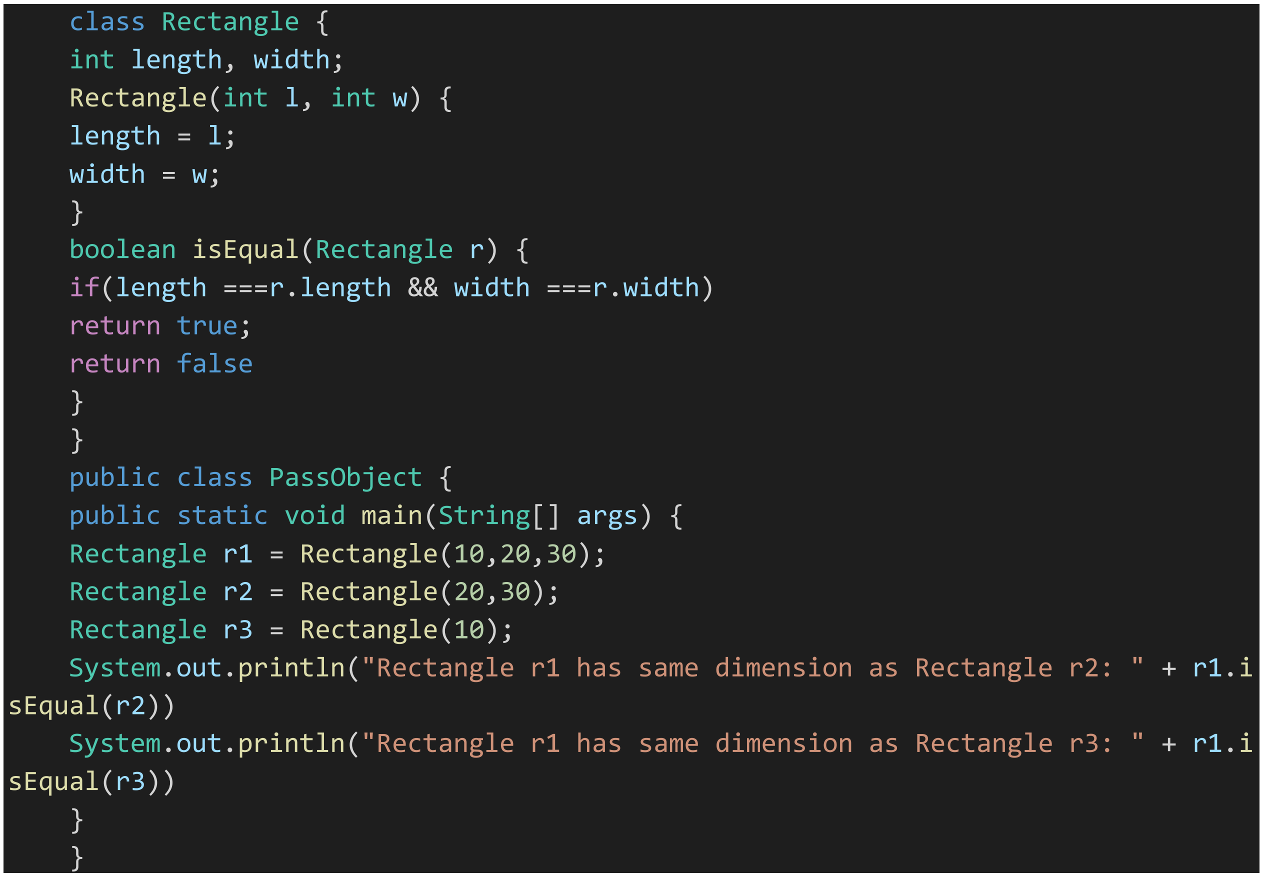
      D) All the above
4. **Match the following**

• Parameterized Constructor	assign()
• Default Constructor	assign(int a)

5. Identify mistakes in code Snippet



```
    class Example {
    public static void main(String args[])
    {
        System.out.println("Multiplication Table of 7");
        int a = 7, ans;

        for (i = 1, i <= 10; i++) {
            ans = a * i
            System.out.println(ans + "\n")
        }
    }
}
```



```
    class Rectangle {
    int length, width;
    Rectangle(int l, int w) {
    length = l;
    width = w;
    }
    boolean isEqual(Rectangle r) {
    if(length ===r.length && width ===r.width)
    return true;
    return false
    }
    }
    public class PassObject {
    public static void main(String[] args) {
    Rectangle r1 = Rectangle(10,20,30);
    Rectangle r2 = Rectangle(20,30);
    Rectangle r3 = Rectangle(10);
    System.out.println("Rectangle r1 has same dimension as Rectangle r2: " + r1.isEqual(r2))
    System.out.println("Rectangle r1 has same dimension as Rectangle r3: " + r1.isEqual(r3))
    }
    }
```

Unit 3 (Inheritance and Interfaces)

1. **Fill in the blanks**
    - The______keyword is used to create an array
    - ______class is mother of all Java classes
2. **True/False**
    - hashcode() returns the address of hash value that is used to search object in a collection. (T/F)
    - When a class implements an interface, it is adding that interface's type to its type. (T/F)
3. **Multiple choice questions**

- Which of these is correct way of inheriting class A by class B?

    a) class B + class A {}

    b) class B inherits class A {}

    c) class B extends A {}

    d) class B extends class A {}

- 2. A class member declared protected becomes a member of subclass of which type?

  a) public member

  b) private member

  c) protected member

  d) static member

4. **Match the following**

• Method Overriding	Compile-Time
• Method Overloading	Run-Time

5. Identify mistakes in code Snippet

```
    class Employee{
    float salary=40000;
   }
   class Programmer extends Employee{
    int bonus=10000;
    public static void main(String args[]){
      Programmer p=new Programmer();
      System.out.println("Programmer salary is:"+salary);
      System.out.println("Bonus of Programmer is:"+bonus);
   }
   }
```

```
class Animal{
    void eat(){System.out.println("eating...");}
    }
    class Dog extends Animal{
    void bark(){System.out.println("barking...");}
    }
    class TestInheritance{
    public static void main(String args[]){
    Dog d=new Dog();
    d.bark();
    d.eat();
    }
    }
```

Unit 4 (Exception Handling)

1. Fill in the blanks

- A package is a pack (group) of_______,_________and other packages.
- Exception Handling is a mechanism to handle_______errors.

2. True/False

- Program statements that you want to monitor for exceptions are contained within a catch block. (T/F)
- finally block will be executed whenever execution leaves a try/catch block, no matter what condition causes it. (T/F)

3. Multiple choice questions

- Which of these keywords is not a part of exception handling?

  a) try

  b) finally

  c) thrown

  d) catch

- Which of this access specifies can be used for a class so that its members can be accessed by a different class in the same package?

  a) Public

  b) Protected

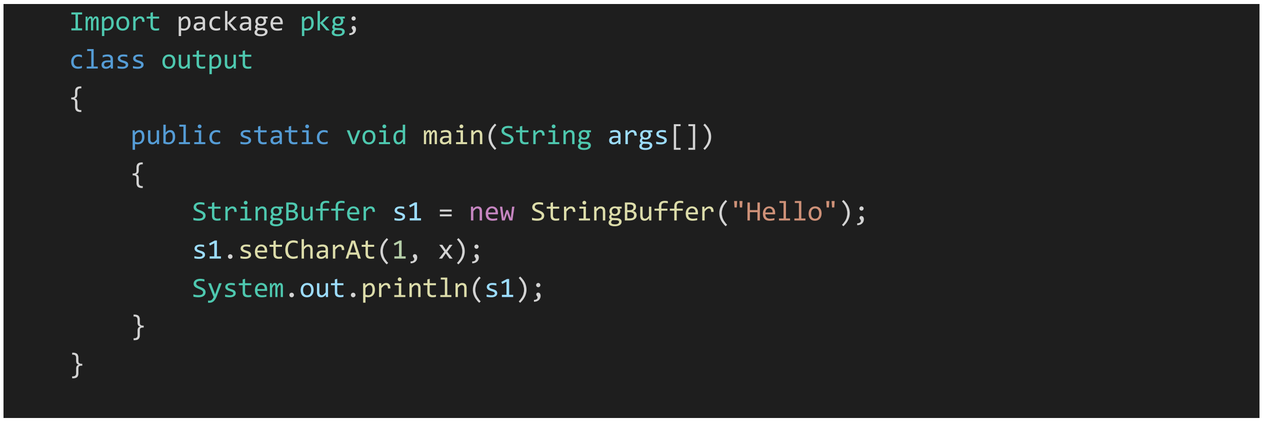
  c) No Modifier

  d) All of the mentioned

4. Match the following

• 1/0	ArrayIndexOutOfBoundsException
• int i=5; for(i=0;i<10;i++)	ArithmeticException

5. Identify mistakes in code Snippet

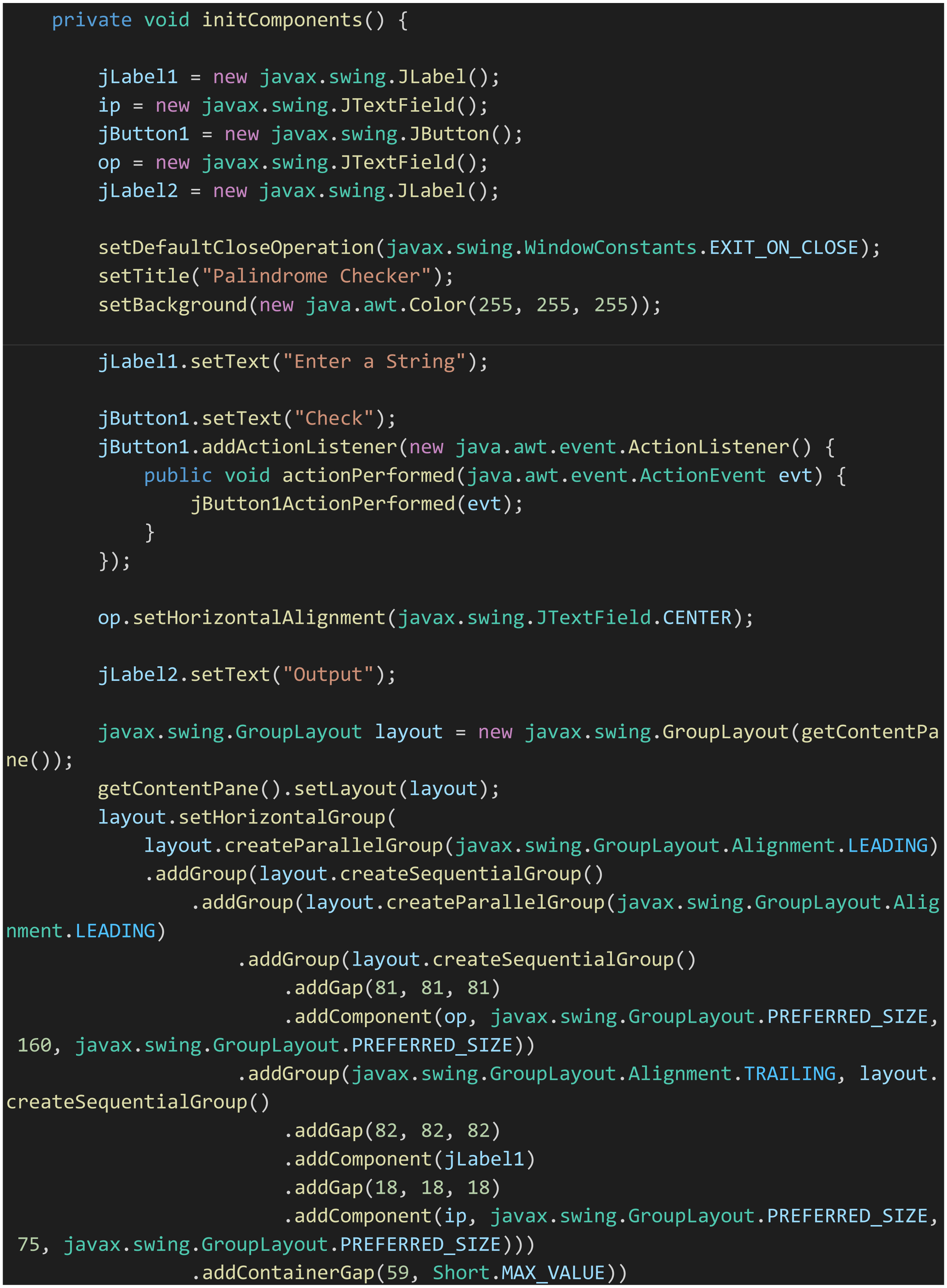


```
Import package pkg;
class output
{
    public static void main(String args[])
    {
        StringBuffer s1 = new StringBuffer("Hello");
        s1.setCharAt(1, x);
        System.out.println(s1);
    }
}
```

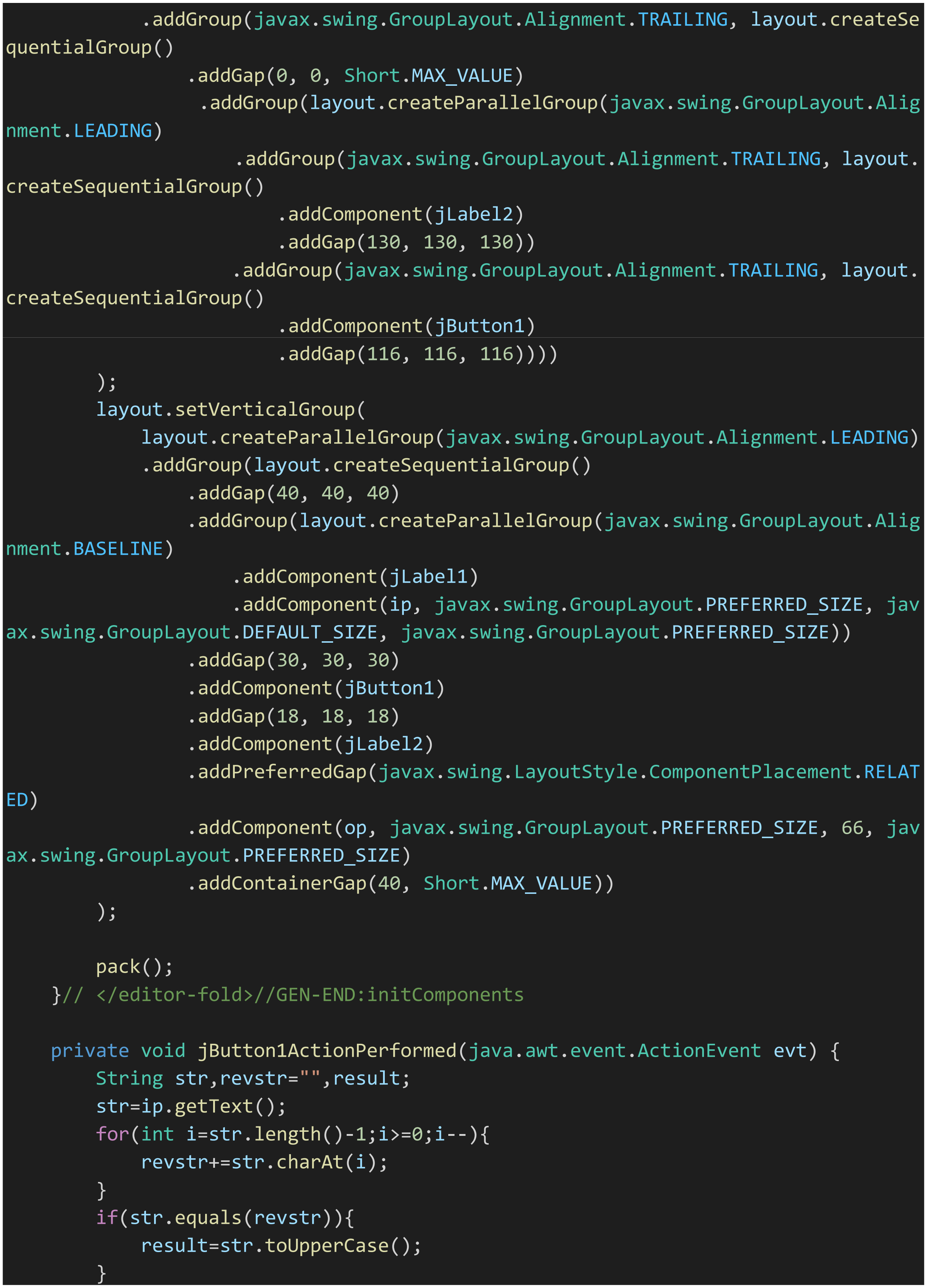
```
    class exception_handling
    {
        public static void main(String args[])
        {
            try
            {
                System.out.print("Hello" + " " + 1 / 0);
            }
            catches(ArithmeticException e)
            {
                System.out.print("World");
            }
        }
    }
```

2) Construct a GUI App using swing to accept a string on click of a button check if it is palindrome and if so, print it in Upper-case in another Text box. If it is not palindrome print the string in reverse order.

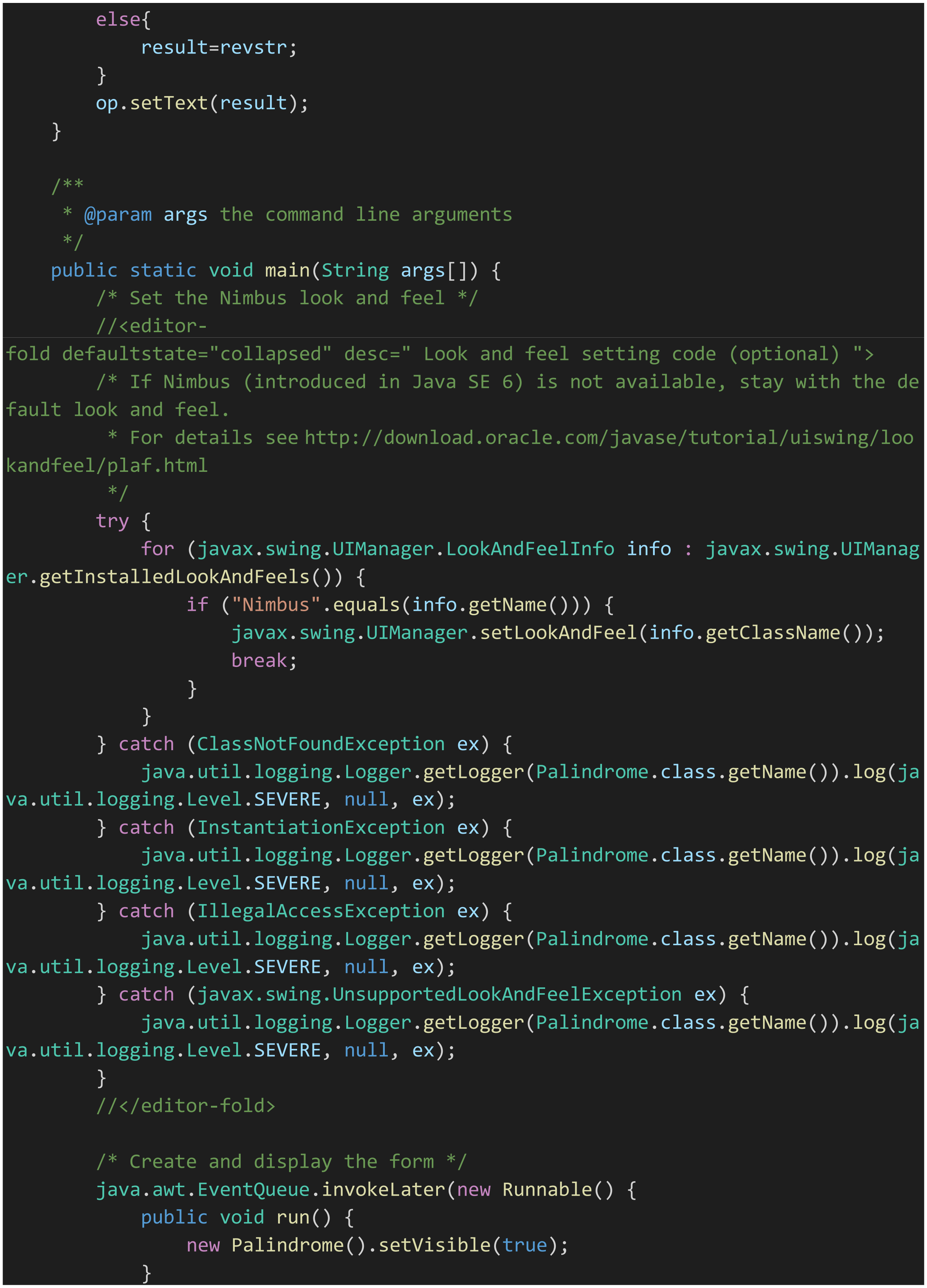
```
package javaapplication2;

/**
 *
 * @author akhil
 */
public class Palindrome extends javax.swing.JFrame {

    /**
     * Creates new form Palindrome
     */
    public Palindrome() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
```



```
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        ip = new javax.swing.JTextField();
        jButton1 = new javax.swing.JButton();
        op = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Palindrome Checker");
        setBackground(new java.awt.Color(255, 255, 255));

        jLabel1.setText("Enter a String");

        jButton1.setText("Check");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });

        op.setHorizontalAlignment(javax.swing.JTextField.CENTER);

        jLabel2.setText("Output");

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPa
ne());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alig
nment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(81, 81, 81)
                        .addComponent(op, javax.swing.GroupLayout.PREFERRED_SIZE,
 160, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.
createSequentialGroup()
                        .addGap(82, 82, 82)
                        .addComponent(jLabel1)
                        .addGap(18, 18, 18)
                        .addComponent(ip, javax.swing.GroupLayout.PREFERRED_SIZE,
 75, javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addContainerGap(59, Short.MAX_VALUE))
```

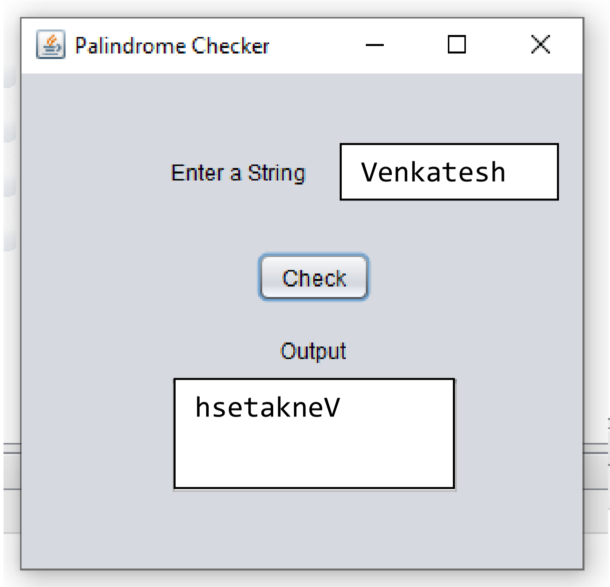


```
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                 .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addComponent(jLabel2)
                        .addGap(130, 130, 130))
                    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                        .addComponent(jButton1)
                        .addGap(116, 116, 116))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(40, 40, 40)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel1)
                    .addComponent(ip, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(30, 30, 30)
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(op, javax.swing.GroupLayout.PREFERRED_SIZE, 66, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(40, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String str,revstr="",result;
        str=ip.getText();
        for(int i=str.length()-1;i>=0;i--){
            revstr+=str.charAt(i);
        }
        if(str.equals(revstr)){
            result=str.toUpperCase();
        }
```



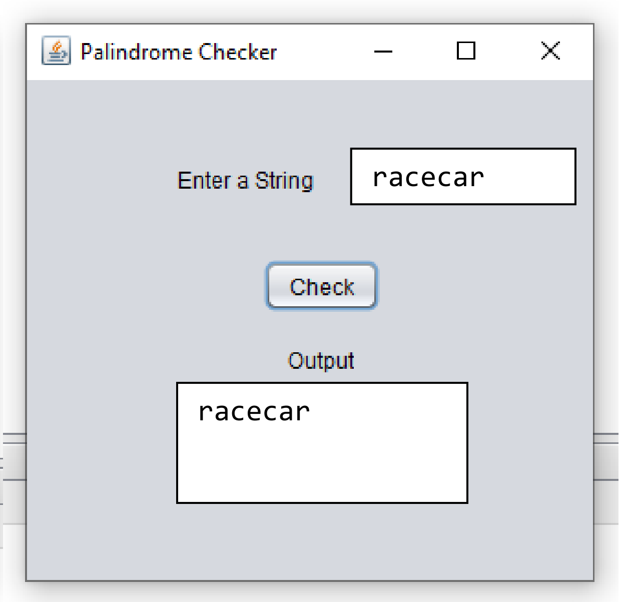
```
        else{
            result=revstr;
        }
        op.setText(result);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.
         * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Palindrome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Palindrome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Palindrome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Palindrome.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Palindrome().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JTextField ip;
    private javax.swing.JButton jButton1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JTextField op;
    // End of variables declaration//GEN-END:variables
}
```

Output:



Enter a String: Venkatesh

Output: hsetakneV



Enter a String: racecar

Output: racecar