Godavari Institute Of Management & Research, Jalgaon

Name: __ _ _ _ _ _ _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **1**

Subject: Lab on Java Programming Sign.of Teacher

Title: Implement a program that demonstrates program structure of java with use of arithmetical and 21 logical implementation.

Objective:

The program demonstrates basic **arithmetic operations** (addition, subtraction, multiplication, division) and **logical operations** (AND, OR, NOT).

Arithmetic Operations:

- The program demonstrates basic arithmetic operations such as addition, subtraction, multiplication, and division with integers.

Logical Operations:

- Logical operations such as AND (`&&`), OR (`||`), and NOT (`!`) are performed using boolean values (`true` and `false`).

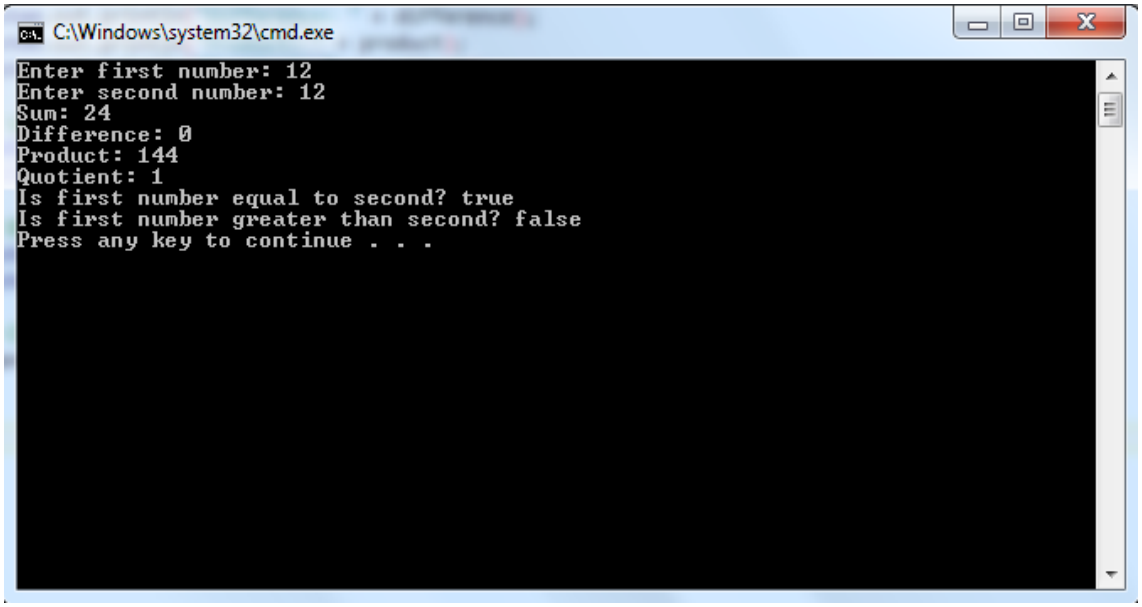
Steps for Java Program:

1. **Declare the Class**
   - Create a class that will hold the program.
2. **Declare the `main` Method**
   - Define the `main` method where execution starts.
3. **Import Required Packages (Optional)**
   - Import any necessary packages (e.g., `Scanner` for user input).
4. **Declare Variables**
   - Declare variables for arithmetic and logical operations (e.g., integers and booleans).
5. **Perform Arithmetic Operations**
   - Use arithmetic operators to perform operations like addition, subtraction, multiplication, division, and modulo.
6. **Perform Logical Operations**
   - Use logical operators (AND, OR, NOT) to perform logical comparisons.
7. **Display Results**
   - Output the results of both arithmetic and logical operations.
8. **Optional: Take User Input (if needed)**
   - Use `Scanner` to allow the user to input values for the operations.
9. **End the Program**
   - End the program after completing all operations and output.

1. **Implement a program that demonstrates program structure of java with use of arithmetical and 21 logical implementation.**

```
import java.util.Scanner;
public class SimpleArithmeticAndLogical {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter first number: ");
      int num1 = scanner.nextInt();
      System.out.print("Enter second number: ");
      int num2 = scanner.nextInt();
      int sum = num1 + num2;
      int difference = num1 - num2;
      int product = num1 * num2;
      int quotient = (num2 != 0) ? num1 / num2 : 0;  // Prevent division by zero
      System.out.println("Sum: " + sum);
      System.out.println("Difference: " + difference);
      System.out.println("Product: " + product);
      System.out.println("Quotient: " + quotient);
          boolean isEqual = (num1 == num2);
      boolean isGreater = (num1 > num2);
      System.out.println("Is first number equal to second? " + isEqual);
      System.out.println("Is first number greater than second? " + isGreater);
      scanner.close();
   }
}
```

OUTPUT:



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **2**

Subject: Lab on Java Programming Sign.of Teacher**:**

Title: Implement a program that demonstrates string operations using String and String Buffer class.

Objective: To demonstrate string operations using the `String` and `StringBuffer` classes in Java, showing how to perform various string manipulations such as concatenation, comparison, and modification using both immutable (`String`) and mutable (`StringBuffer`) string objects.

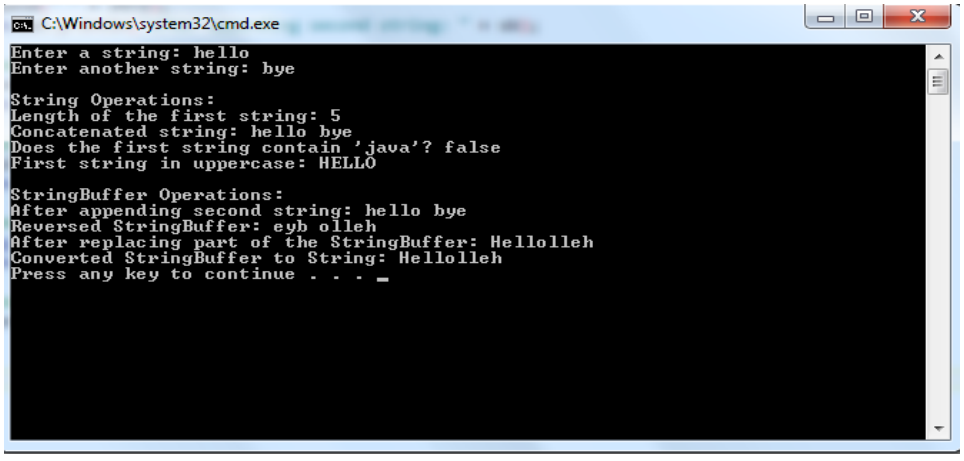
Steps for Java Program:

1. **Declare the Class**
    - Define a class, for example, StringBufferDemo, which will hold the string operations.
2. **Declare the main Method**
    - Define the main method as the entry point of the program.
3. **String Operations Using the String Class**
    - Create String objects for string manipulations.
    - Perform the following operations:
        - **Concatenation** using + or concat() method.
        - **Equality check** using equals() method.
        - **Case conversion** using toUpperCase() and toLowerCase().
        - **Length** using length().
        - **Substring** extraction using substring().
        - **Comparison** using compareTo().
4. **String Operations Using the StringBuffer Class**
    - Create StringBuffer objects for mutable string manipulations.
    - Perform the following operations:
        - **Append** using append() method.
        - **Insert** using insert() method.
        - **Reverse** using reverse() method.
        - **Replace** parts of the string using replace() method.
        - **Delete** characters using delete() or deleteCharAt().
5. **Display Results**
    - Use System.out.println() to display the results of all the string operations on the console.
6. **End the Program**
    - Close the main method and end the program execution.

Implement a program that demonstrates string operations using String and String Buffer class.

```
import java.util.Scanner;
public class StringBufferStringOperations {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a string: ");
      String str = scanner.nextLine();
      System.out.print("Enter another string: ");
      String str2 = scanner.nextLine();
      System.out.println("\nString Operations:");
      System.out.println("Length of the first string: " + str.length());
      String combined = str + " " + str2;
      System.out.println("Concatenated string: " + combined);
      boolean containsWord = str.contains("java");
      System.out.println("Does the first string contain 'java'? " + containsWord);
      String upperStr = str.toUpperCase();
      System.out.println("First string in uppercase: " + upperStr);
      StringBuffer sb = new StringBuffer(str);
      System.out.println("\nStringBuffer Operations:");
      sb.append(" " + str2);
      System.out.println("After appending second string: " + sb);
      sb.reverse();
      System.out.println("Reversed StringBuffer: " + sb);
      sb.replace(0, 5, "Hello");
      System.out.println("After replacing part of the StringBuffer: " + sb);
      String convertedStr = sb.toString();
      System.out.println("Converted StringBuffer to String: " + convertedStr);
      scanner.close();
   }
}
```

OUTPUT:



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **3**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrates inner class and static fields.

Objective: To demonstrate the use of **inner classes** and **static fields** in Java. This will show how an inner class can be defined inside an outer class and how static fields can be used to store class-level data.

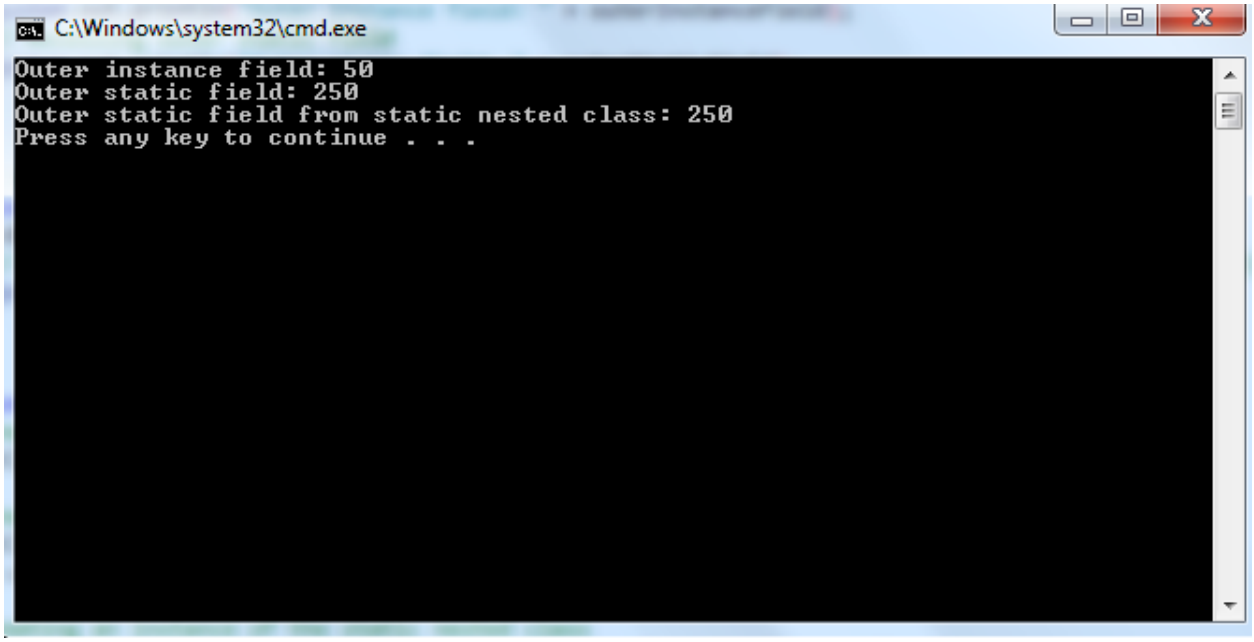
Steps for Java Program:

1. **Declare the Outer Class**
   - Create the outer class that will contain both the static field and the inner class.
2. **Declare the Static Field**
   - Inside the outer class, declare a static field (variable) that belongs to the class rather than an instance of the class.
3. **Declare the Inner Class**
   - Inside the outer class, define the inner class. The inner class can be non-static or static.
4. **Access Static Field from Inner Class**
   - Inside the inner class, demonstrate how to access the static field of the outer class.
5. **Create an Instance of the Outer Class in the main Method**
   - In the main method, create an instance of the outer class to demonstrate its functionality.
6. **Instantiate the Inner Class**
   - Instantiate the inner class either from an instance of the outer class (for non-static inner class) or directly using the outer class (for static inner class).
7. **Display Results**
   - Use System.out.println() to display the static field's value and any other outputs to the console.
8. **End the Program**
   - Complete the main method and close the program.

Implement a program that demonstrates inner class and static fields.

```
public class OuterClass {
    static int outerStaticField = 250;
    int outerInstanceField = 50;
    class InnerClass {
        void display() {
            System.out.println("Outer instance field: " + outerInstanceField);
            System.out.println("Outer static field: " + outerStaticField);
        }
    }
    static class StaticNestedClass {
        void display() {
            System.out.println("Outer static field from static nested class: " +
outerStaticField);
        }
    }
    public static void main(String[] args) {
            OuterClass outer = new OuterClass();
        OuterClass.InnerClass inner = outer.new InnerClass();
        inner.display(); // This will print fields from the outer class
        OuterClass.StaticNestedClass staticNested = new OuterClass.StaticNestedClass();
        staticNested.display(); // This will print the static field from the outer class
    }
}
```

OUTPUT:-



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **4**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrate inheritance, polymorphism

Objective: To demonstrate **inheritance** and **polymorphism** in Java. Inheritance allows one class to inherit fields and methods from another class, while polymorphism enables a method to behave differently based on the object it is acting upon.

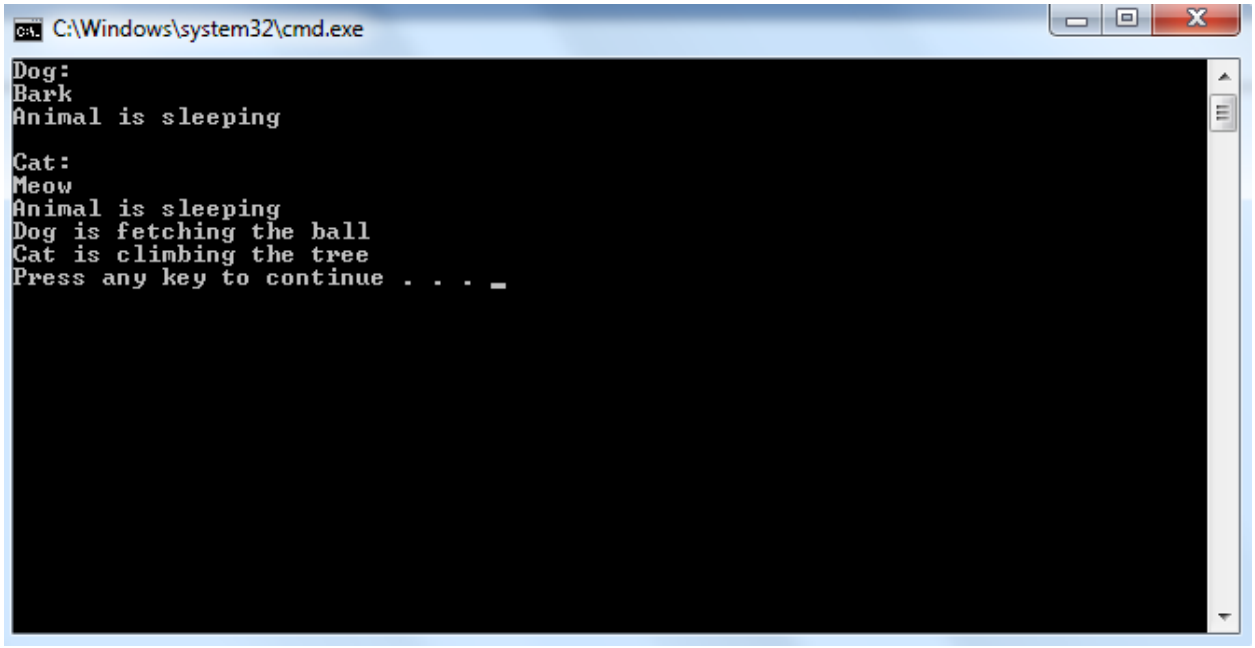
Steps for Java Program:

1. **Declare the Parent (Base) Class**
   - Define a parent class that will contain common fields and methods.
2. **Declare Inherited Methods**
   - In the parent class, declare methods that can be inherited by child classes.
3. **Declare the Child (Subclass) Class**
   - Define a child class that extends the parent class, inheriting its properties and behaviors.
4. **Override Methods in the Child Class (Polymorphism)**
   - In the child class, override the methods from the parent class to provide specific behavior (demonstrating polymorphism).
5. **Create Instances in the `main` Method**
   - In the `main` method, create objects of the parent class and the child class to demonstrate inheritance and polymorphism.
6. **Use Method Overriding to Demonstrate Polymorphism**
   - Call the overridden methods on instances of both parent and child classes and observe polymorphism (same method behaving differently depending on the object).
7. **Display Results**
   - Use `System.out.println()` to display the results of calling methods on both parent and child class objects.
8. **End the Program**
   - Complete the `main` method and close the program.

Implement a program that demonstrate inheritance, polymorphism

```
class Animal {
    public void sound() {
        System.out.println("Some animal sound");
    }
    public void sleep() {
        System.out.println("Animal is sleeping");
    }
}
class Dog extends Animal {
    @Override
    public void sound() {
        System.out.println("Bark");
    }
    public void fetch() {
        System.out.println("Dog is fetching the ball");
    }
}
class Cat extends Animal {
    @Override
    public void sound() {
        System.out.println("Meow");
    }
    public void climb() {
        System.out.println("Cat is climbing the tree");
    }
}
public class InheritancePolymorphismDemo {
    public static void main(String[] args) {
        Animal myDog = new Dog();
        Animal myCat = new Cat();
        System.out.println("Dog:");
        myDog.sound();
        myDog.sleep();
        System.out.println("\nCat:");
        myCat.sound();
        myCat.sleep();
        if (myDog instanceof Dog) {
            Dog dog = (Dog) myDog;
            dog.fetch();
        }
        if (myCat instanceof Cat) {
            Cat cat = (Cat) myCat;
            cat.climb();
        }
    }
}
```

OUTPUT:-


C:\Windows\system32\cmd.exe
Dog:
Bark
Animal is sleeping
Cat:
Meow
Animal is sleeping
Dog is fetching the ball
Cat is climbing the tree
Press any key to continue . . .

Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **5**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrates 2D shapes on frames.

Objective: To demonstrate the creation and display of **2D shapes** (such as rectangles, circles, and lines) on a **frame** in Java. This will involve using **Swing** for the graphical user interface (GUI) and **Graphics** to draw shapes.

Steps for Java Program:

1. **Import Required Packages**
   - Import necessary packages for creating a GUI (`javax.swing.*`) and for drawing graphics (`java.awt.*`).
2. **Create the Frame (JFrame)**
   - Define a class that extends `JFrame` to create a window (frame) where the shapes will be displayed.
3. **Override the `paint()` Method**
   - Override the `paint(Graphics g)` method to handle the drawing of 2D shapes on the frame.
   - Use the `Graphics` object to draw shapes such as circles, rectangles, and lines.
4. **Create and Display Shapes**
   - Inside the `paint()` method, use the following methods to draw shapes:
     - **Draw Rectangle** using `g.drawRect(x, y, width, height).`
     - **Draw Oval (Circle)** using `g.drawOval(x, y, width, height).`
     - **Draw Line** using `g.drawLine(x1, y1, x2, y2).`
5. **Create the `main` Method**
   - In the `main` method, create an instance of the frame and set its properties, such as size, visibility, and close operation.
   - Instantiate the class that extends `JFrame` to show the window.
6. **Set Frame Properties**
   - Set the size of the frame using `frame.setSize(width, height).`
   - Set the frame to be visible using `frame.setVisible(true).`
   - Optionally, set the default close operation using `frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).`
7. **Display the Frame**
   - When the `main` method runs, the frame with the drawn shapes will appear on the screen.
8. **End the Program**
   - End the program after setting the frame properties and displaying the shapes.

Implement a program that demonstrates 2D shapes on frames.

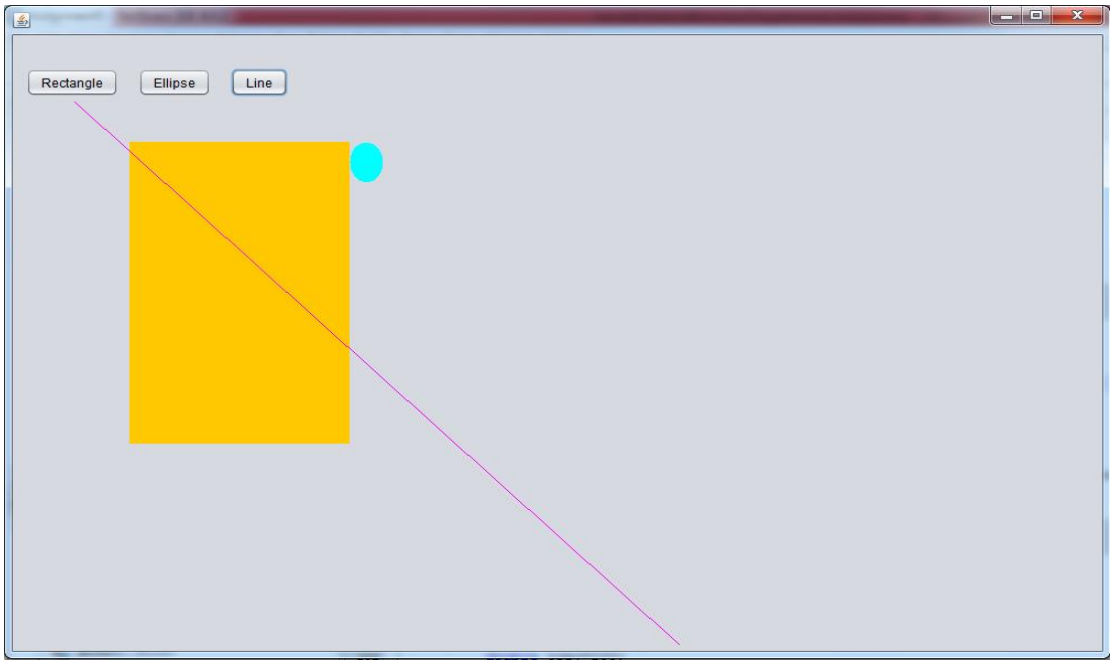
```
import java.awt.*;
import java.awt.geom.*;
public class NewJFrame extends javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
    }
     @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jButton1.setText("Rectangle");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jButton2.setText("Ellipse");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });
        jButton4.setText("Line");
        jButton4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton4ActionPerformed(evt);
            }
        });
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jButton1)
                .addGap(18, 18, 18)
                .addComponent(jButton2)
                .addGap(18, 18, 18)
                .addComponent(jButton4)
                .addContainerGap(92, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(27, 27, 27)
```

```
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASEL
INE)
                .addComponent(jButton1)
                .addComponent(jButton2)
                .addComponent(jButton4))
            .addContainerGap(228, Short.MAX_VALUE))
        );
        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(59, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );

        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
Graphics g1=jPanel1.getGraphics();
        Graphics2D g2  = (Graphics2D)g1;
      g2.setPaint(Color.ORANGE);
        double leftx=100;
        double topy=100;
        double width=100;
      double height=200;//For Squre width and height should be same
        Rectangle2D rect = new Rectangle2D.Double(leftx,topy,leftx+width,topy+height);
      g2.fill(rect);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
Graphics g1=jPanel1.getGraphics();
        Graphics2D g2  = (Graphics2D)g1;
      g2.setPaint(Color.CYAN);
        double leftx=300;
        double topy=100;
        double width=30;
      double height=40;//For Circle width and height should be same
```

```
        Ellipse2D ellipse = new Ellipse2D.Double(leftx,topy,width,height);
      g2.fill(ellipse);
    }
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         Graphics g1=jPanel1.getGraphics();
        Graphics2D g2  = (Graphics2D)g1;
      g2.setPaint(Color.MAGENTA);
        double startx=50;
        double starty=60;
        double endx=600;
      double endy=600;
        Line2D line = new Line2D.Double(startx,starty,endx,endy);
      g2.draw(line);
    }
    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(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.
SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.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 NewJFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton4;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration
}
```

OUTPUT:



Godavari Institute Of Management & Research, Jalgaon

Name: _ — _ _ _ _ _ _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **6**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrates color and fonts.

Objective: To demonstrate how to use **colors** and **fonts** in Java when drawing text and shapes on a **frame**. This will involve using **Swing** and **Graphics** for customizing the color and font styles in a graphical user interface.

Steps for Java Program:

1. **Import Required Packages**
   - Import necessary packages:
     - javax.swing.* for creating the window (frame).
     - java.awt.* for working with graphics, colors, and fonts.
2. **Create the Frame (JFrame)**
   - Define a class that extends JFrame to create the main window where graphics will be displayed.
3. **Override the paint() Method**
   - Override the paint(Graphics g) method to enable custom drawing.
   - Inside this method, the Graphics object will be used for drawing and styling.
4. **Set the Color**
   - Use the Graphics.setColor(Color color) method to set the drawing color.
   - Create Color objects to define colors (e.g., Color.RED, Color.BLUE, new Color(255, 0, 0)).
5. **Set the Font**
   - Use the Graphics.setFont(Font font) method to set the font for drawing text.
   - Create a Font object with the desired font family, style, and size (e.g., new Font("Arial", Font.BOLD, 20)).
6. **Draw Shapes with Color**
   - Use the Graphics methods (e.g., drawRect(), drawOval(), fillRect(), fillOval()) to draw shapes in the selected color.
7. **Draw Text with Color and Font**
   - Use the Graphics.drawString(String str, int x, int y) method to draw text with the selected font and color.
8. **Create the main Method**
   - In the main method, create an instance of the class that extends JFrame and set its properties like size, visibility, and default close operation.
9. **Set Frame Properties**
   - Set the size of the frame using frame.setSize(width, height).
   - Set the frame to be visible using frame.setVisible(true).
   - Optionally, set the default close operation using frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).
10. **Display the Frame**
    - When the main method is executed, the frame with colored shapes and text in the selected fonts will appear.

11. **End the Program**
    - Complete the main method, marking the end of the program.

Implement a program that demonstrates color and fonts.

```
import java.awt.*;
import java.awt.geom.*;
import java.util.*;
public class NewJFrame extends javax.swing.JFrame {
   public NewJFrame() {
      initComponents();
     }
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {
      jPanel1 = new javax.swing.JPanel();
      jButton1 = new javax.swing.JButton();
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
      jButton1.setText("Click");
      jButton1.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
         }
      });
      javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
      jPanel1.setLayout(jPanel1Layout);
      jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(213, 213, 213)
            .addComponent(jButton1)
            .addContainerGap(667, Short.MAX_VALUE))
      );
      jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jButton1)
            .addGap(0, 578, Short.MAX_VALUE))
      );
```

```
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(20, 20, 20)
                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(55, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(22, 22, 22)
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        GraphicsEnvironment
ge=GraphicsEnvironment.getLocalGraphicsEnvironment();
            String s[]=ge.getAvailableFontFamilyNames();
        Graphics g1=jPanel1.getGraphics();
        Random rd = new Random();
        int y=50;
        int sz=20;
                    for(int i=0;i<s.length;i++)
                    {
                            Font f=new Font(s[i],Font.BOLD,sz);//Font.ITALIC
                            g1.setFont(f);
                    int r=rd.nextInt(255);
                    int g=rd.nextInt(255);
                    int b=rd.nextInt(255);
                    Color c=new Color(r,g,b);
                    g1.setColor(c);

                            g1.drawString("Hello World",50,y);
                    y=y+20;
```

```
                sz=sz+1;
            }
    }
    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(NewJFrame.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
        }
        //</editor-fold>

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

```
    });
  }
  // Variables declaration - do not modify
  private javax.swing.JButton jButton1;
  private javax.swing.JPanel jPanel1;
  // End of variables declaration
}
```

OUTPUT:

Godavari Institute Of Management & Research, Jalgaon

Name: __ _ _ _ _ _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **7**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program to illustrate use of various swing components.

Objective: To demonstrate the use of various **Swing components** in Java, including buttons, labels, text fields, checkboxes, radio buttons, combo boxes, and panels to create a basic graphical user interface (GUI).

Steps for Java Program:

1. **Import Required Packages**
   - Import necessary Swing and AWT packages:
     - javax.swing.* for Swing components (e.g., JButton, JLabel, JTextField).
     - java.awt.* for layout management and basic AWT components.
2. **Create the Main Frame (JFrame)**
   - Define a class that extends JFrame to create the main window for the GUI.
3. **Set Layout Manager**
   - Set the layout manager for the frame (e.g., FlowLayout, GridLayout, BorderLayout) to control component placement.
4. **Create Swing Components**
   - Create various Swing components such as:
     - **JButton**: Create buttons to trigger actions.
     - **JLabel**: Create labels to display text.
     - **JTextField**: Create text fields for user input.
     - **JCheckBox**: Create checkboxes for options.
     - **JRadioButton**: Create radio buttons for selecting a single option from a set.
     - **ButtonGroup**: Group radio buttons together so only one can be selected at a time.
     - **JComboBox**: Create combo boxes (drop-down lists) for multiple choices.
5. **Add Components to Frame**
   - Add the created components to the frame using add() method.
6. **Add Event Handling (Optional)**
   - Attach action listeners to components such as buttons, checkboxes, and radio buttons to handle user interactions.
   - Use addActionListener() for buttons or ItemListener for checkboxes and radio buttons to capture events.
7. **Set Frame Properties**
   - Set the size of the frame using frame.setSize(width, height).
   - Set the frame to be visible using frame.setVisible(true).
   - Optionally, set the default close operation using frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).
8. **Display the Frame**
   - When the main method is executed, the frame with all the Swing components will appear.

9. **End the Program**
    - Complete the main method and close the program.

Implement a program to illustrate use of various swing components.

```
package assignment7;
public class NewJFrame extends javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        buttonGroup1 = new javax.swing.ButtonGroup();
        jPanel1 = new javax.swing.JPanel();
        jLabel1 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        jTextArea1 = new javax.swing.JTextArea();
        jLabel3 = new javax.swing.JLabel();
        jCheckBox1 = new javax.swing.JCheckBox();
        jCheckBox2 = new javax.swing.JCheckBox();
        jCheckBox3 = new javax.swing.JCheckBox();
        jButton1 = new javax.swing.JButton();
        jLabel4 = new javax.swing.JLabel();
        jRadioButton1 = new javax.swing.JRadioButton();
        jRadioButton2 = new javax.swing.JRadioButton();
        jLabel5 = new javax.swing.JLabel();
        jComboBox1 = new javax.swing.JComboBox();
        jLabel6 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        jList1 = new javax.swing.JList();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jLabel1.setText("Enter Rno");
        jLabel2.setText("Enter Name");
        jTextArea1.setColumns(20);
        jTextArea1.setRows(5);
        jScrollPane1.setViewportView(jTextArea1);
        jLabel3.setText("Favorite Color");
        jCheckBox1.setText("Red");
        jCheckBox2.setText("Green");
        jCheckBox3.setText("Blue");
        jButton1.setText("Click");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
```

```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
        jLabel4.setText("Class");
        buttonGroup1.add(jRadioButton1);
        jRadioButton1.setText("MCA-1");
        buttonGroup1.add(jRadioButton2);
        jRadioButton2.setText("MCA-2");
        jLabel5.setText("Laptop");
        jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"HP", "Dell", "Lenovo" }));
        jLabel6.setText("Subject");
        jList1.setModel(new javax.swing.AbstractListModel() {
            String[] strings = { "C", "C++", "Java" };
            public int getSize() { return strings.length; }
            public Object getElementAt(int i) { return strings[i]; }
        });
        jScrollPane2.setViewportView(jList1);
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(35, 35, 35)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TR
AILING, false)
                            .addComponent(jLabel6,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, 62, Short.MAX_VALUE)
                            .addComponent(jLabel5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(44, 44, 44)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                            .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                    .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 92,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                .addGroup(jPanel1Layout.createSequentialGroup()
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 68,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 96,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(25, 25, 25)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                            .addComponent(jCheckBox1)
                            .addComponent(jCheckBox2)
                            .addComponent(jCheckBox3)
                            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 146,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 89,
javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(13, 13, 13)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                            .addComponent(jRadioButton1)
                            .addComponent(jRadioButton2))))))
            .addContainerGap(691, Short.MAX_VALUE))
        );
```

```
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(55, 55, 55)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 29,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 62,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 25,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(12, 12, 12)
                        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE,
35, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(18, 18, 18)
                        .addComponent(jCheckBox1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jCheckBox2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                        .addComponent(jCheckBox3)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addGap(21, 21, 21)
                        .addComponent(jRadioButton1)
```

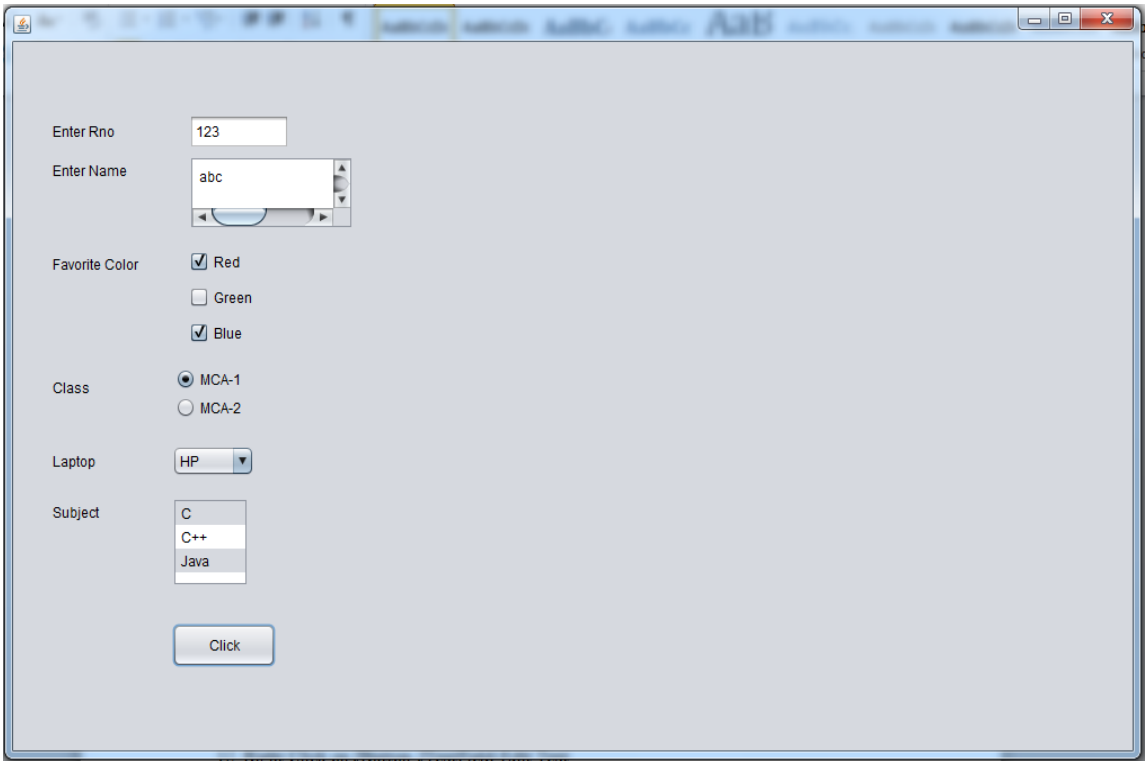
```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jRadioButton2)
                    .addGap(23, 23, 23))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE,
25, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(36, 36, 36)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TR
AILING)
                .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jComboBox1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jScrollPane2,
javax.swing.GroupLayout.PREFERRED_SIZE, 75,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(31, 31, 31)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(64, Short.MAX_VALUE))
        );

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addGap(0, 0, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        pack();
    }// </editor-fold>
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        System.out.println("Rno= "+jTextField1.getText());
        System.out.println("Name= "+jTextArea1.getText());
        String color=" ";
        if (jCheckBox1.isSelected())
        color=color+" "+jCheckBox1.getText();
        if (jCheckBox2.isSelected())
        color=color+" "+jCheckBox2.getText();
        if (jCheckBox3.isSelected())
        color=color+" "+jCheckBox3.getText();
        System.out.println("Favorite Colors= "+color);
        String cl=" ";
        if (jRadioButton1.isSelected())
        cl=cl+" "+jRadioButton1.getText();
        else
        cl=cl+" "+jRadioButton2.getText();
        System.out.println("Class= "+cl);
        System.out.println("Laptop= "+jComboBox1.getSelectedItem().toString());
        System.out.println("Subjects= ");
        Object o[]=jList1.getSelectedValues();
        for(int i=0;i<o.length;i++)
        {
            System.out.println(o[i].toString());
        }

    }
    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(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        }
        //</editor-fold>

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

    // Variables declaration - do not modify
```

```
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JButton jButton1;
    private javax.swing.JCheckBox jCheckBox1;
    private javax.swing.JCheckBox jCheckBox2;
    private javax.swing.JCheckBox jCheckBox3;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JList jList1;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JScrollPane jScrollPane2;
    private javax.swing.JTextArea jTextArea1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

OUTPUT:



Godavari Institute Of Management & Research, Jalgaon

Name: _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ __ _ _ _ _ _ _ Roll No: _ _ _ __

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **8**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrates use of dialog box and menus.

Objective: To demonstrate the use of **dialog boxes** and **menus** in Java using **Swing**. Dialog boxes are used to display information or request input from the user, while menus provide an interactive interface for users to choose from various actions.

Steps for Java Program:

1. **Import Required Packages**
    - Import necessary packages:
        - javax.swing.* for Swing components (e.g., JFrame, JMenuBar, JMenu, JMenuItem, JOptionPane).
        - java.awt.* for layout and event handling.
2. **Create the Main Frame (JFrame)**
    - Define a class that extends JFrame to create the main window for the GUI.
3. **Create a Menu Bar**
    - Create a JMenuBar to hold the menus.
    - Create menus (e.g., JMenu) and menu items (e.g., JMenuItem) for the menu options.
    - Add event listeners to menu items to define actions (e.g., show dialog boxes, exit the application).
4. **Add the Menu Bar to the Frame**
    - Use setJMenuBar() to add the created menu bar to the frame.
5. **Create Dialog Boxes**
    - Use JOptionPane to show different types of dialog boxes:
        - **Message Dialog**: Display a simple message using JOptionPane.showMessageDialog().
        - **Input Dialog**: Request user input using JOptionPane.showInputDialog().
        - **Confirm Dialog**: Ask for a confirmation (Yes/No) using JOptionPane.showConfirmDialog().
6. **Handle Menu Actions**
    - Attach action listeners to menu items to trigger dialog boxes or other actions.
    - For example, when a user selects a menu item, show a message dialog, input dialog, or confirmation dialog based on the action.
7. **Set Frame Properties**
    - Set the size of the frame using frame.setSize(width, height).
    - Set the frame to be visible using frame.setVisible(true).
    - Optionally, set the default close operation using frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).
8. **Display the Frame**
    - When the main method is executed, the frame with the menu and dialog boxes will appear.
9. **End the Program**
    - Complete the main method and close the program.

Implement a program that demonstrates use of dialog box and menus.

```
package assignment8;
import javax.swing.*;
import java.io.*;
import java.awt.*;
public class NewJFrame extends javax.swing.JFrame {
    public NewJFrame() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPopupMenu1 = new javax.swing.JPopupMenu();
        Red = new javax.swing.JMenuItem();
        Green = new javax.swing.JMenuItem();
        Blue = new javax.swing.JMenuItem();
        jDialog1 = new javax.swing.JDialog();
        jTextField1 = new javax.swing.JTextField();
        Click = new javax.swing.JButton();
        jPanel1 = new javax.swing.JPanel();
        jMenuBar1 = new javax.swing.JMenuBar();
        jMenu1 = new javax.swing.JMenu();
        jMenuItem1 = new javax.swing.JMenuItem();
        jSeparator1 = new javax.swing.JPopupMenu.Separator();
        jMenuItem2 = new javax.swing.JMenuItem();
        jSeparator2 = new javax.swing.JPopupMenu.Separator();
        jCheckBoxMenuItem1 = new javax.swing.JCheckBoxMenuItem();
        jSeparator3 = new javax.swing.JPopupMenu.Separator();
        jRadioButtonMenuItem1 = new javax.swing.JRadioButtonMenuItem();
        jSeparator5 = new javax.swing.JPopupMenu.Separator();
        jMenuItem6 = new javax.swing.JMenuItem();
        jSeparator4 = new javax.swing.JPopupMenu.Separator();
        jMenuItem4 = new javax.swing.JMenuItem();
        jMenu2 = new javax.swing.JMenu();
        jMenuItem3 = new javax.swing.JMenuItem();
        Red.setText("Red");
        Red.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                RedActionPerformed(evt);
            }
        });
        jPopupMenu1.add(Red);
```

```
        Green.setText("Green");
        Green.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                GreenActionPerformed(evt);
            }
        });
        jPopupMenu1.add(Green);
        Blue.setText("Blue");
        Blue.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                BlueActionPerformed(evt);
            }
        });
        jPopupMenu1.add(Blue);
        jDialog1.getContentPane().setLayout(new java.awt.FlowLayout());
        jTextField1.setText("jTextField1");
        jDialog1.getContentPane().add(jTextField1);
        Click.setText("Click");
        jDialog1.getContentPane().add(Click);
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jPanel1.setComponentPopupMenu(jPopupMenu1);
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 958, Short.MAX_VALUE)
        );
        jPanel1Layout.setVerticalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 581, Short.MAX_VALUE)
        );

        jMenu1.setText("File");
        jMenu1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenu1ActionPerformed(evt);
            }
        });


    jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEve
    nt.VK_A, java.awt.event.InputEvent.CTRL_MASK));
```

```
        jMenuItem1.setText("InputDialogBox");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem1);
        jMenu1.add(jSeparator1);


jMenuItem2.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEve
nt.VK_B, java.awt.event.InputEvent.CTRL_MASK));
        jMenuItem2.setText("MessageDialogBox");
        jMenuItem2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem2ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem2);
        jMenu1.add(jSeparator2);
jCheckBoxMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.eve
nt.KeyEvent.VK_C, java.awt.event.InputEvent.ALT_MASK));
        jCheckBoxMenuItem1.setSelected(true);
        jCheckBoxMenuItem1.setText("ConfirmDialogBox");
        jCheckBoxMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jCheckBoxMenuItem1ActionPerformed(evt);
            }
        });
        jMenu1.add(jCheckBoxMenuItem1);
        jMenu1.add(jSeparator3);
jRadioButtonMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.e
vent.KeyEvent.VK_D, java.awt.event.InputEvent.SHIFT_MASK));
        jRadioButtonMenuItem1.setSelected(true);
        jRadioButtonMenuItem1.setText("OptionDialogBox");
        jRadioButtonMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jRadioButtonMenuItem1ActionPerformed(evt);
            }
        });
        jMenu1.add(jRadioButtonMenuItem1);
        jMenu1.add(jSeparator5);
        jMenuItem6.setText("FileChooser");
        jMenuItem6.addActionListener(new java.awt.event.ActionListener() {
```

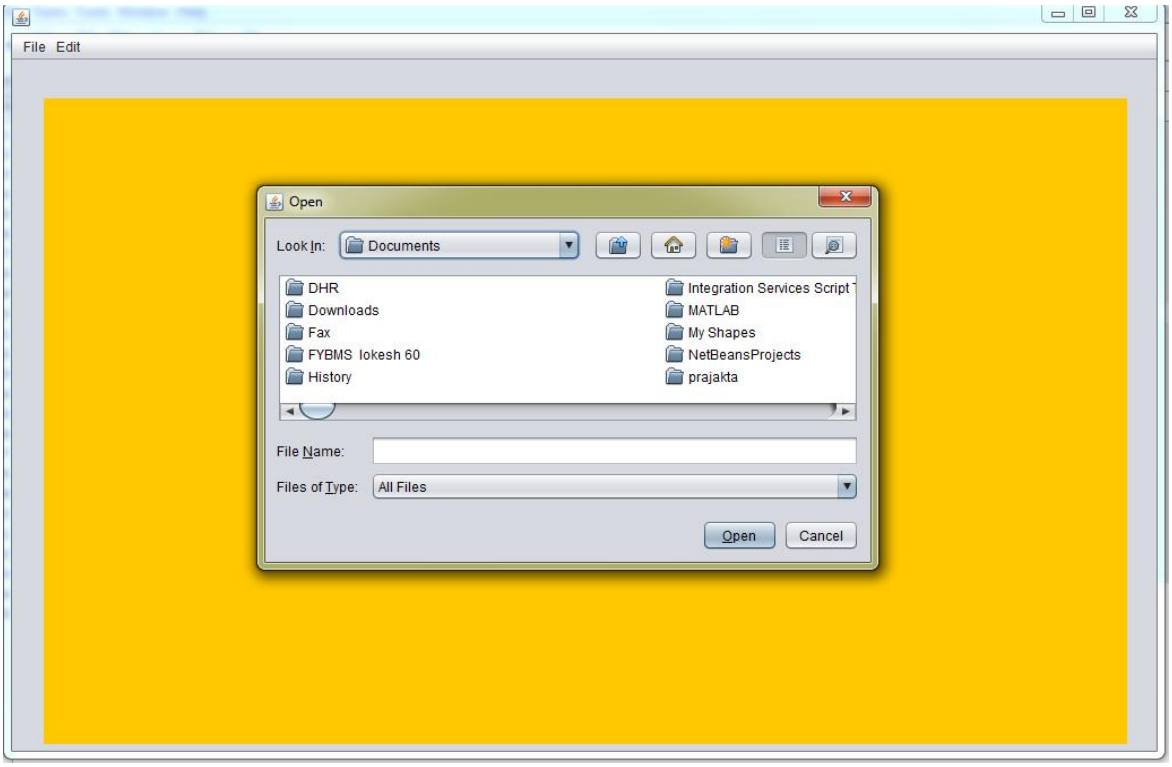
```
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem6ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem6);
        jMenu1.add(jSeparator4);

        jMenuItem4.setText("ColorChooser");
        jMenuItem4.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem4ActionPerformed(evt);
            }
        });
        jMenu1.add(jMenuItem4);
        jMenuBar1.add(jMenu1);
        jMenu2.setText("Edit");
        jMenuItem3.setText("UserDialogBox");
        jMenuItem3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem3ActionPerformed(evt);
            }
        });
        jMenu2.add(jMenuItem3);
        jMenuBar1.add(jMenu2);
        setJMenuBar(jMenuBar1);
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(28, 28, 28)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(26, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(35, 35, 35)
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String n=JOptionPane.showInputDialog("Enter Name");
        System.out.println("Name="+n);
    }
    private void jMenuItem3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jDialog1.setTitle("This is my DialogBox");
        jDialog1.setSize(222,222);
        jDialog1.show();
    }
    private void jMenuItem2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JOptionPane.showMessageDialog(null,"Success");
    }
    private void jCheckBoxMenuItem1ActionPerformed(java.awt.event.ActionEvent evt)
{
        // TODO add your handling code here:
        int i=JOptionPane.showConfirmDialog(null, "Are you Sure?");
        System.out.println(i);
    }
    private void jRadioButtonMenuItem1ActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
        String[] options = {"first", "second", "third"};
        int x = JOptionPane.showOptionDialog(null, "Select Option",
                "OptionDialogBox",JOptionPane.DEFAULT_OPTION,
JOptionPane.INFORMATION_MESSAGE, null, options, options[0]);
        System.out.println("Your Option is "+x);
    }
    private void RedActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jPanel1.setBackground(Color.red);
    }
    private void GreenActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        jPanel1.setBackground(Color.green);
    }
    private void BlueActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        jPanel1.setBackground(Color.blue);
    }
    private void jMenu1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    private void jMenuItem6ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
JFileChooser fc=new JFileChooser();
        int i=fc.showOpenDialog(this);
        if(i==JFileChooser.APPROVE_OPTION)
        {
             File f=fc.getSelectedFile();
            String filepath=f.getPath();
            System.out.println("You Selected "+filepath);
        }
    }
    private void jMenuItem4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Color c=JColorChooser.showDialog(this,"Select a color",Color.ORANGE);
        jPanel1.setBackground(c);
    }
    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(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JMenuItem Blue;
    private javax.swing.JButton Click;
    private javax.swing.JMenuItem Green;
    private javax.swing.JMenuItem Red;
    private javax.swing.JCheckBoxMenuItem jCheckBoxMenuItem1;
    private javax.swing.JDialog jDialog1;
    private javax.swing.JMenu jMenu1;
    private javax.swing.JMenu jMenu2;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JMenuItem jMenuItem2;
    private javax.swing.JMenuItem jMenuItem3;
    private javax.swing.JMenuItem jMenuItem4;
    private javax.swing.JMenuItem jMenuItem6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPopupMenu jPopupMenu1;
    private javax.swing.JRadioButtonMenuItem jRadioButtonMenuItem1;
    private javax.swing.JPopupMenu.Separator jSeparator1;
    private javax.swing.JPopupMenu.Separator jSeparator2;
    private javax.swing.JPopupMenu.Separator jSeparator3;
    private javax.swing.JPopupMenu.Separator jSeparator4;
    private javax.swing.JPopupMenu.Separator jSeparator5;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

Output:


File Edit
Open
Look In:
Documents
DHR
Downloads
Fax
FYBMS lokesh 60
History
Integration Services Script
MATLAB
My Shapes
NetBeansProjects
prajakta
File Name:
Files of Type:
All Files
Open
Cancel

Godavari Institute Of Management & Research, Jalgaon

Name: _— _ _ _ _ _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **9**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrates event handling for various types of events.

Objective: To demonstrate **event handling** in Java for various types of events, such as action events, mouse events, key events, and window events, using **Swing** components. This will show how to capture and respond to user actions like button clicks, mouse movements, keyboard input, and window interactions.

Steps for Java Program:

1. **Import Required Packages**
   - Import necessary packages:
     - javax.swing.* for GUI components (e.g., JButton, JFrame).
     - java.awt.* and java.awt.event.* for handling events (e.g., ActionListener, MouseListener, KeyListener, WindowListener).
2. **Create the Main Frame (JFrame)**
   - Define a class that extends JFrame to create the main window for the GUI.
3. **Add Swing Components**
   - Add components like buttons (JButton), labels (JLabel), text fields (JTextField), etc., to the frame for triggering events.
4. **Implement Action Event Handling**
   - Use ActionListener to handle button clicks and other actions.
   - Implement actionPerformed() method to define the behavior when an action occurs (e.g., a button click).
5. **Implement Mouse Event Handling**
   - Use MouseListener to handle mouse events (e.g., mouse clicks, mouse entered, mouse exited).
   - Implement methods like mouseClicked(), mouseEntered(), mouseExited(), mousePressed(), and mouseReleased() to define actions for mouse events.
6. **Implement Key Event Handling**
   - Use KeyListener to handle keyboard input.
   - Implement methods like keyPressed(), keyReleased(), and keyTyped() to define actions based on key events.
7. **Implement Window Event Handling**
   - Use WindowListener to handle window-related events (e.g., window opening, closing, etc.).
   - Implement methods like windowOpened(), windowClosing(), windowClosed(), etc., to define actions for window events.
8. **Attach Event Listeners to Components**
   - Attach the appropriate event listener to each component (e.g., addActionListener() for buttons, addMouseListener() for mouse events, addKeyListener() for key events).
9. **Set Frame Properties**
   - Set the size of the frame using frame.setSize(width, height).
   - Set the frame to be visible using frame.setVisible(true).

- Optionally, set the default close operation using frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE).

10. **Display the Frame**
    - When the main method is executed, the frame will appear and interact with the user through various event handlers.
11. **End the Program**
    - Complete the main method and close the program.

Implement a program that demonstrates event handling for various types of events.

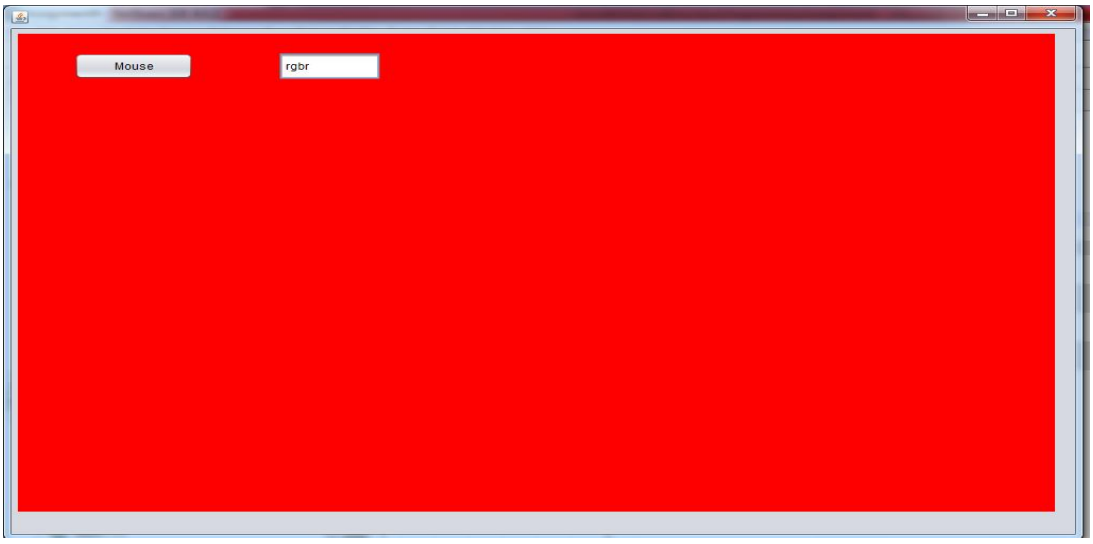
```
package assignment9;
import java.awt.Color;
public class NewJFrame extends javax.swing.JFrame {
   public NewJFrame() {
      initComponents();
   }
   @SuppressWarnings("unchecked")
   // <editor-fold defaultstate="collapsed" desc="Generated Code">
   private void initComponents() {
      jPanel1 = new javax.swing.JPanel();
      jButton2 = new javax.swing.JButton();
      jTextField1 = new javax.swing.JTextField();
      setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
      jPanel1.addMouseListener(new java.awt.event.MouseAdapter() {
         public void mouseClicked(java.awt.event.MouseEvent evt) {
            jPanel1MouseClicked(evt);
         }
      });
      jButton2.setText("Mouse");
      jButton2.addMouseListener(new java.awt.event.MouseAdapter() {
         public void mouseEntered(java.awt.event.MouseEvent evt) {
            jButton2MouseEntered(evt);
         }
         public void mouseExited(java.awt.event.MouseEvent evt) {
            jButton2MouseExited(evt);
         }
      });
      jTextField1.addKeyListener(new java.awt.event.KeyAdapter() {
         public void keyTyped(java.awt.event.KeyEvent evt) {
            jTextField1KeyTyped(evt);
         }
      });
      javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
      jPanel1.setLayout(jPanel1Layout);
      jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(53, 53, 53)
```

```
            .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 112,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(81, 81, 81)
            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
95, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(635, Short.MAX_VALUE))
        );
        jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(24, 24, 24)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
33, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(541, Short.MAX_VALUE))
        );
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(19, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(28, Short.MAX_VALUE))
        );
        pack();
    }// </editor-fold>
    private void jButton2MouseEntered(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        jPanel1.setBackground(Color.red);
    }
    private void jButton2MouseExited(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
```

```
        jPanel1.setBackground(Color.GREEN);
    }
    private void jTextField1KeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        char a=evt.getKeyChar();
                        if(a=='r' || a=='R')
                        {
                                jPanel1.setBackground(Color.red);

                        }
                else if(a=='g' || a=='G')
                        {
                          jPanel1.setBackground(Color.GREEN);

                        }
                else
                {
                      jPanel1.setBackground(Color.BLACK);
                }
    }
 int count=0;
    private void jPanel1MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
        count++;
        if(count==1)
            jPanel1.setBackground(Color.RED);
        else if(count==2)
            jPanel1.setBackground(Color.GREEN);
        else if(count==3)
            jPanel1.setBackground(Color.BLUE);
        else
            count=0;
    }
    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(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    // End of variables declaration
}
```

Output:



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **10**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program to illustrate multithreading.

Objective: To demonstrate the concept of multithreading in Java by creating multiple threads that execute concurrently to perform different tasks simultaneously.

Steps:

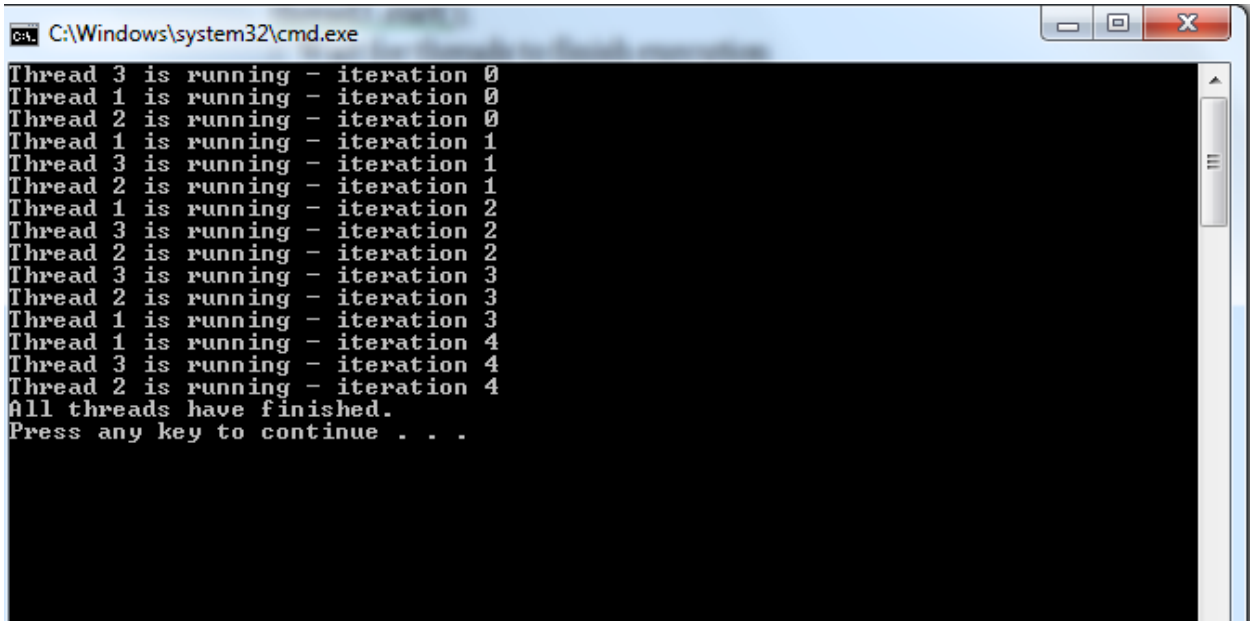
1. **Create a class that implements the `Runnable` interface**.
   - Define a `run()` method to specify the task that the thread will execute.
2. **Create a `Thread` object**.
   - Instantiate a `Thread` object and pass the `Runnable` implementation to it.
3. **Override the `run()` method** in the `Runnable` implementation.
   - Include the task logic you want each thread to perform.
4. **Start the threads** using the `start()` method.
   - This will invoke the `run()` method in each thread concurrently.
5. **Use `Thread.sleep()` (optional)** to simulate delays and demonstrate concurrency.
6. **(Optional) Synchronize threads** if accessing shared resources to avoid data inconsistency.
7. **Monitor the threads** by printing outputs in the `run()` method to observe parallel execution.

Implement a program to illustrate multithreading.

```
class MyThread extends Thread {
    private int threadNumber;
    public MyThread(int threadNumber) {
        this.threadNumber = threadNumber;
    }
    @Override
    public void run() {
        for (int i = 0; i < 5; i++) {
            try {
                // Simulate some work with sleep
                Thread.sleep(500);
                System.out.println("Thread " + threadNumber + " is running - iteration " +
i);
            } catch (InterruptedException e) {
                System.out.println("Thread " + threadNumber + " was interrupted.");
            }
        }
    }
```

```
}
public class MultithreadingExample {
    public static void main(String[] args) {
        // Create and start multiple threads
        MyThread thread1 = new MyThread(1);
        MyThread thread2 = new MyThread(2);
        MyThread thread3 = new MyThread(3);
        // Start the threads
        thread1.start();
        thread2.start();
        thread3.start();
        // Wait for threads to finish execution
        try {
            thread1.join();
            thread2.join();
            thread3.join();
        } catch (InterruptedException e) {
            System.out.println("Main thread was interrupted.");
        }
        System.out.println("All threads have finished.");
    }
}
```

Output:



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **11**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program to illustrate exception handling.

Objective: To demonstrate exception handling in Java by creating a program that handles runtime and compile-time exceptions, ensuring the program can continue running even when an error occurs.

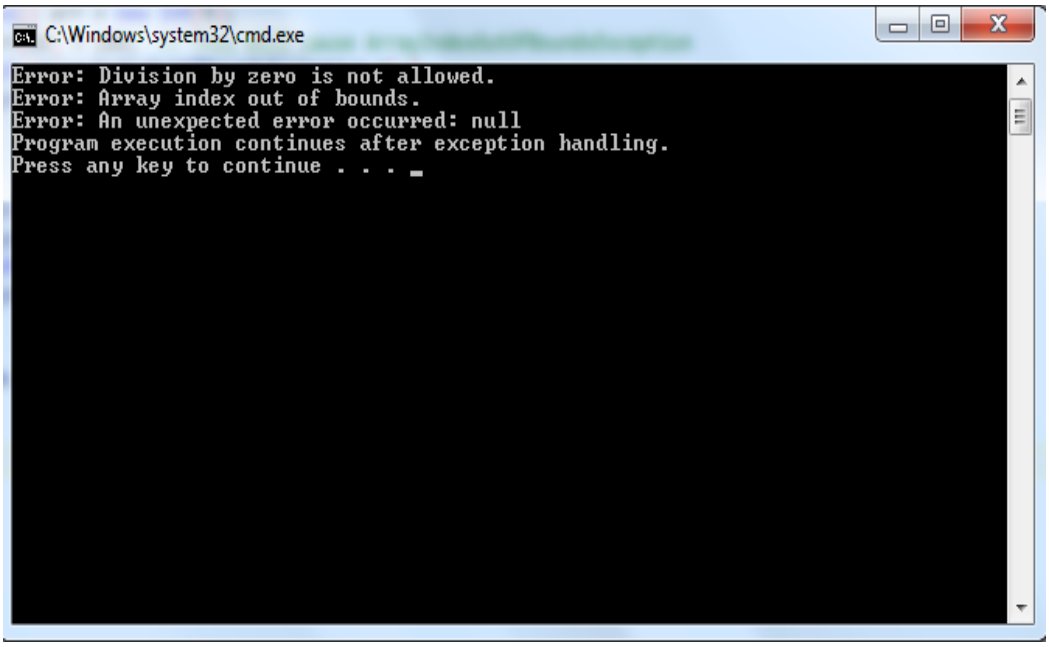
Steps:

1. **Identify the potential areas where exceptions might occur**.
   - Think of operations that can cause exceptions, such as file reading, database access, array indexing, division by zero, etc.
2. **Use `try-catch` block** to handle exceptions:
   - Wrap the code that might throw an exception in the `try` block.
   - Use `catch` blocks to handle specific exceptions.
3. **Handle different types of exceptions**:
   - Use multiple `catch` blocks for different types of exceptions (e.g., `ArithmeticException`, `ArrayIndexOutOfBoundsException`, etc.).
   - Optionally, use a generic `Exception` to catch any unforeseen exceptions.
4. **Use `finally` block** (optional but recommended):
   - The `finally` block will execute regardless of whether an exception was thrown or not (e.g., for cleanup operations like closing files or releasing resources).
5. **Throw exceptions explicitly** (optional):
   - Use `throw` to manually throw exceptions when necessary (e.g., custom exception handling).
6. **Test with various exceptions**:
   - Test scenarios where exceptions occur, such as invalid user input, dividing by zero, etc.
7. **Log or print error messages**:
   - Inside the `catch` block, log or print meaningful messages to help identify the problem.
8. **Program termination**:
   - Ensure the program can either recover or gracefully terminate when an exception is handled.

Implement a program to illustrate exception handling.

```
public class ExceptionHandlingExample {
    public static void main(String[] args) {

        try {
            int result = 10 / 0;  // This will cause ArithmeticException
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }

        try {
            int[] arr = new int[5];
            arr[10] = 100;  // This will cause ArrayIndexOutOfBoundsException
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Error: Array index out of bounds.");
        }

        try {
            String text = null;
            System.out.println(text.length());  // This will cause NullPointerException
        } catch (Exception e) {
            System.out.println("Error: An unexpected error occurred: " + e.getMessage());
        }

        System.out.println("Program execution continues after exception handling.");
    }
}
```

Output:



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no:**12**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program to demonstrate use of File class.

Objective: To demonstrate the use of the `File` class in Java for performing basic file operations such as creating, reading, writing, and deleting files.

Steps:

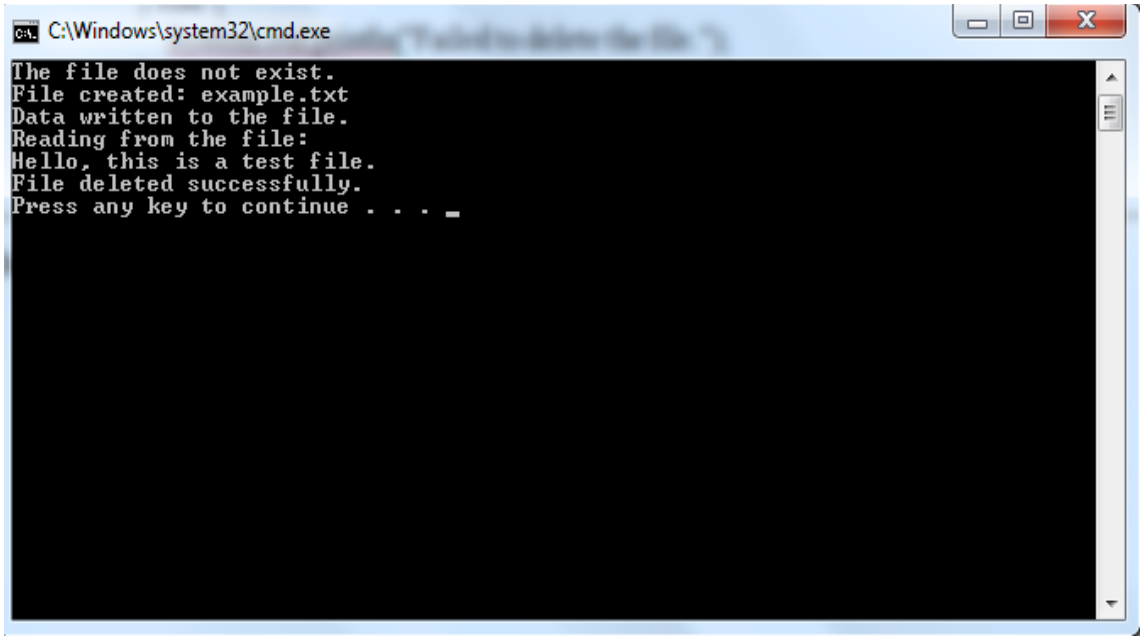
1. **Import the java.io.File class**.
   - Ensure you have the necessary import statement: import java.io.File;
2. **Create a File object**.
   - Instantiate a File object by providing the file path (relative or absolute) as a string.
3. **Check if the file exists**:
   - Use the exists() method to check if a file already exists at the specified location.
4. **Create a new file**:
   - Use the createNewFile() method to create a new file if it doesn't already exist.
5. **Check if the file is a directory or a file**:
   - Use isDirectory() and isFile() methods to check if the File object represents a directory or a regular file.
6. **Write data to a file**:
   - Use classes like FileWriter or BufferedWriter to write text data to the file.
7. **Read data from a file**:
   - Use FileReader or BufferedReader to read the content from the file.
8. **Delete a file**:
   - Use the delete() method to delete the file from the filesystem.
9. **Check file permissions**:
   - Use methods like canRead(), canWrite(), and canExecute() to check file access permissions.
10. **List files in a directory**:
    - Use the list() method to get the names of all files in a directory.
11. **Use length() to get file size**:
    - Call the length() method to get the size of the file in bytes.
12. **Ensure proper exception handling**:
    - Handle IOException using try-catch blocks to manage any IO-related issues during file operations.

Implement a program to demonstrate use of File class.

```
import java.io.File;
import java.io.FileWriter;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;
public class FileClassExample {
   public static void main(String[] args) {
      // Specify the file path
      String filePath = "example.txt";
      // Create a File object
      File file = new File(filePath);
      // Example 1: Check if the file exists
      if (file.exists()) {
         System.out.println("The file already exists.");
      } else {
         System.out.println("The file does not exist.");
         try {
            // Example 2: Create the file
            if (file.createNewFile()) {
               System.out.println("File created: " + file.getName());
            } else {
               System.out.println("File already exists.");
            }
         } catch (IOException e) {
            System.out.println("An error occurred while creating the file.");
            e.printStackTrace();
         }
      }

      // Example 3: Write data to the file
      try (FileWriter writer = new FileWriter(file)) {
         writer.write("Hello, this is a test file.");
         System.out.println("Data written to the file.");
      } catch (IOException e) {
         System.out.println("An error occurred while writing to the file.");
         e.printStackTrace();
      }
      // Example 4: Read data from the file
      try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
         String line;
         System.out.println("Reading from the file:");
         while ((line = reader.readLine()) != null) {
            System.out.println(line);
         }
      } catch (IOException e) {
         System.out.println("An error occurred while reading the file.");
         e.printStackTrace();
```

```
        }

        // Example 5: Delete the file
        if (file.delete()) {
            System.out.println("File deleted successfully.");
        } else {
            System.out.println("Failed to delete the file.");
        }
    }
}
```

Output:



Godavari Institute Of Management & Research, Jalgaon

Name: __ _ _ _ _ _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **13**

Subject: Lab on Java Programming Sign. of Teacher:

Title: Implement a program that demonstrates JDBC on application.

Objective: To demonstrate the use of JDBC (Java Database Connectivity) for connecting to a database, executing SQL queries, and performing basic database operations such as inserting, updating, and retrieving data in Java.

Steps to Demonstrate JDBC in Java:

1. **Import JDBC Packages**:
   - Import java.sql.* for necessary JDBC classes.
2. **Load and Register the JDBC Driver**:
   - Use Class.forName() to load the database driver.
3. **Establish a Database Connection**:
   - Use DriverManager.getConnection() to connect to the database.
4. **Create a Statement Object**:
   - Use Connection.createStatement() or Connection.prepareStatement() to create a Statement or PreparedStatement.
5. **Execute SQL Queries**:
   - Use executeQuery() for SELECT queries.
   - Use executeUpdate() for INSERT, UPDATE, or DELETE queries.
6. **Process the Result Set** (for SELECT queries):
   - Use ResultSet to iterate through and retrieve data.
7. **Handle Data Modification Queries** (INSERT/UPDATE/DELETE):
   - Use executeUpdate() to perform data modification operations.
8. **Use PreparedStatement** (Optional):
   - Use PreparedStatement for parameterized queries to prevent SQL injection.
9. **Close Resources**:
   - Close ResultSet, Statement, and Connection to release resources.
10. **Handle Exceptions**:
    - Use try-catch blocks to handle SQLException.
11. **Use Transactions** (Optional):
    - Use setAutoCommit(false), commit(), and rollback() for handling transactions.

Implement a program that demonstrates JDBC on application.

```
package assignment13;
    import java.sql.*;
    public class NewJFrame extends javax.swing.JFrame {
       public NewJFrame() {
          initComponents();
       }
       @SuppressWarnings("unchecked")
       // <editor-fold defaultstate="collapsed" desc="Generated Code">
       private void initComponents() {

          jPanel1 = new javax.swing.JPanel();
          jLabel1 = new javax.swing.JLabel();
          jTextField1 = new javax.swing.JTextField();
          jLabel2 = new javax.swing.JLabel();
          jTextField2 = new javax.swing.JTextField();
          jButton1 = new javax.swing.JButton();
          jButton2 = new javax.swing.JButton();
          jButton3 = new javax.swing.JButton();
          jButton4 = new javax.swing.JButton();
          setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

          jLabel1.setText("RNo");

          jLabel2.setText("Name");

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

          jButton2.setText("Update");
          jButton2.addActionListener(new java.awt.event.ActionListener() {
             public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
             }
          });

          jButton3.setText("Delete");
          jButton3.addActionListener(new java.awt.event.ActionListener() {
             public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
             }
          });

          jButton4.setText("Select");
          jButton4.addActionListener(new java.awt.event.ActionListener() {
             public void actionPerformed(java.awt.event.ActionEvent evt) {
```

```
            jButton4ActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(83, 83, 83)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)
                .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE, 72,
Short.MAX_VALUE)
                .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE,
83, javax.swing.GroupLayout.PREFERRED_SIZE)
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE,
81, javax.swing.GroupLayout.PREFERRED_SIZE)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                    .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE,
89, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 106,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addContainerGap(569, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(56, 56, 56)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE))
```

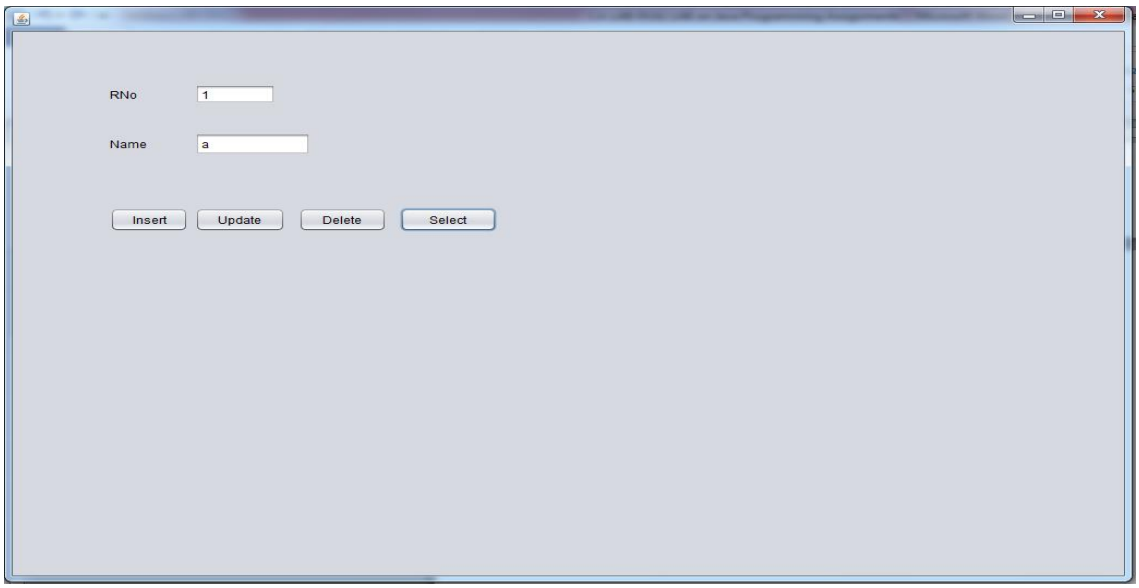
```
                .addGap(33, 33, 33)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                    .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(62, 62, 62)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING, false)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
                        .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addContainerGap(362, Short.MAX_VALUE))
        );
        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(42, Short.MAX_VALUE))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
try
```

```
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       Connection c=DriverManager.getConnection("jdbc:odbc:dsn1"," "," ");
        Statement st=c.createStatement();;
        String s1=jTextField1.getText();
        int i=Integer.parseInt(s1);
        String s2=jTextField2.getText();
        int count=st.executeUpdate("insert into student values("+i+",'"+s2+"')");
        System.out.println("Record Inserted "+count);
        }
      catch(Exception e)
        {
          System.out.println("Insert Exp "+e);
        }
  }
  private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
try
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       Connection c=DriverManager.getConnection("jdbc:odbc:dsn1"," "," ");
        Statement st=c.createStatement();;
        String s1=jTextField1.getText();
        int i=Integer.parseInt(s1);
        String s2=jTextField2.getText();
        int count=st.executeUpdate("update student set sname='"+s2+"' where
rno="+i+"");
        System.out.println("Record Updated "+count);
        }
      catch(Exception e)
        {
          System.out.println("Update Exp "+e);
        }
  }
  private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
try
        {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       Connection c=DriverManager.getConnection("jdbc:odbc:dsn1"," "," ");
        Statement st=c.createStatement();;
        String s1=jTextField1.getText();
        int i=Integer.parseInt(s1);
        int count=st.executeUpdate("delete * from student where rno="+i+"");
        System.out.println("Record Deleted "+count);
        }
      catch(Exception e)
        {
          System.out.println("Delete Exp "+e);
        }
  }
```

```
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try
          {
          Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
          Connection c=DriverManager.getConnection("jdbc:odbc:dsn1"," "," ");
          Statement st=c.createStatement();;
          String s1=jTextField1.getText();
          int i=Integer.parseInt(s1);
          ResultSet rs=st.executeQuery("select * from student where rno="+i+"");
          while(rs.next())
          {
                jTextField2.setText(rs.getString("sname"));
           }
          }
        catch(Exception e)
           {
              System.out.println("Select Exp "+e);
           }

    }

    /**
     * @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(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (InstantiationException ex) {
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
```

```
java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Le
vel.SEVERE, null, ex);
        }
        //</editor-fold>
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new NewJFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}
```

OUTPUT:-



Godavari Institute Of Management & Research, Jalgaon

Name: _ Roll No: _ _ _ _ _

Date of Performance: _ _ /_ _/20_ _ Batch: _ _ _ _ _ _

Class: M.C.A. (I) Practical no: **14**

Subject: Lab on Java Programming Sign. of Teacher**:**

Title: Implement a program that demonstrate package creation and use in program.

Objective: To demonstrate how to create a package, organize classes into packages, and use them in a Java program.

Steps:

1. **Create a directory structure for the package**:
   - Create a directory with the desired package name, e.g., com/example/utility/.
2. **Create a class inside the package**:
   - Inside the package directory, create a Java class with a package declaration at the top.
   - Example: com/example/utility/Greeting.java.
3. **Write code inside the class**:
   - Add methods and functionality to the class.
   - Example: A sayHello method inside Greeting class.
4. **Compile the class inside the package**:
   - Use javac with the -d option to specify the destination for compiled classes.
   - Command: javac -d . com/example/utility/Greeting.java.
5. **Create another class in a different package or default package (Main program)**:
   - Write a class that will use the class from the created package.
   - Use the import statement to access the class from the package.
6. **Compile the main class**:
   - Compile the class that uses the imported class.
   - Command: javac Main.java.
7. **Run the main program**:
   - Use the java command to run the main class.
   - Command: java Main.
8. **Verify the output**:
   - Check the console output to see if the package and class were used correctly.

Implement a program that demonstrate package creation and use in program.

```
package assignment14;
import mypackage.NewClass;
public class Assignment14
{
    public static void main(String[] args)
    {
        NewClass n=new NewClass();
        n.show();
    }
  }
//Create mypackage, Create NewClass
package mypackage;
public class NewClass
{
    public void show()
    {
        System.out.println("Show Method is Called");
    }
}
```

OUTPUT:-

```
Show Method is Called
```