Practical 1: Write a program to find out a factorial of as given number.

```
package class_obj;
import class_obj.Facto;
import java.util.*;
class Facto     {
public static void main(String args[])     {
int n,fact=1,i;
System.out.println("Enter the Number");
Scanner r=new Scanner(System.in);
n=r.nextInt();
for(i=1;i<=n;i++)   {
fact=fact*i;     }
System.out.println("Factorial of the number is: " +fact);    }
}
```

Output:

Enter the Number

6

Factorial of the number is: 720

Practical 2: Write a program to demonstrate any 2 types of operators used in java.

```
package class_obj;
import class_obj.operators;
import java.util.*;
public class operators    {
public static void main(String args[])       {
Scanner s=new Scanner(System.in);
System.out.println("Arithmatic Operators");
System.out.println("Enter First Number:");
int num1=s.nextInt();
System.out.println("Enter Second Number:");
int num2=s.nextInt();

//Arithematic Operators
int sum=num1+num2;
        System.out.println("Sum of the two numbers:" +sum);
int product=num1*num2;
System.out.println("Product of the two numbers:" +product);

//Relational Operators
 int a,b;
```

```
System.out.println("Enter two numbers");
Scanner r=new Scanner(System.in);
a=r.nextInt();
b=r.nextInt();
System.out.println("Relational Operators");
System.out.println("true/false" +(a<b));
System.out.println("true/false" +(a>b));
System.out.println("true/false" +(a<=b));
System.out.println("true/false" +(a>=b));
System.out.println("true/false" +(a==b));
System.out.println("true/false" +(a!=b));
}
}
```

Output:

```
Arithmatic Operators
Enter First Number:
10
Enter Second Number:
20
Sum of the two numbers:30
Product of the two numbers:200
Enter two numbers
30
```

10

Relational Operators

true/falsefalse

true/falsetrue

true/falsefalse

true/falsetrue

true/falsefalse

true/falsetrue

Practical 3: Write a program to print a Fibonacci series up to given terms.

```
package class_obj;
import class_obj.Fibo;
import java.util.*;
class Fibo        {
public static void main(String args[])       {
int term, a=0, b=1, c,i;
System.out.println("Enter the term :");
Scanner r=new Scanner(System.in);
term=r.nextInt();
for(i=1;i<=term;i++)    {
System.out.print(a+"\n"+  " ");
c=a+b;
a=b;
b=c;        }  }  }
```

Output:

```
Enter the term :
10
0
 1
 1
 2
 3
```

5

8

13

21

34

Practical 4: Write a program to implement different types of string methods.

```
package class_obj;
import class_obj.string_functions;
import java.util.*;
class string_functions    {
public static void main(String args[])     {
String a="HELLO";
String b="Java Programming";
System.out.println(a.toLowerCase());
System.out.println(b.toUpperCase());
System.out.println(a.concat(b));

System.out.println( "Length of the string is :"+a.length());

String c="Aachal";
System.out.println(c.trim());

System.out.println(c.charAt(2));

System.out.println(c.indexOf('l'));
System.out.println(b.equals(a));

String d="Gita";
```

```
        System.out.println(d.replace('G','S'));
        System.out.println(d.isEmpty());  }     }
```

Output:

```
hello
JAVA PROGRAMMING
HELLOJava Programming
Length of the string is :5
Aachal
c
5
false
Sita
False
```

Practical 5: Write a program to print a given number is prime or not.

```
package class_obj;
import class_obj.PrimeNumber;
import java.util.*;
class PrimeNumber      {
public static void main (String args[])     {
int n,count=0;
System.out.println("Enter the Number:");
Scanner s=new Scanner(System.in);
n=s.nextInt();         for(int i=1;i<=n;i++)        {
if(n%i==0)    {
count++;    }  }
if(count == 2)      {
System.out.println("Prime Number");
else {
System.out.println("Not Prime Number");    }     }     }
```

Output:

Enter the Number:

2

Prime Number

Enter the Number:

10

Not Prime Number

Practical 6: Write a program to demonstrate a use of command line argument.

```
package Program;
public class Java_practicals            {
public static void main(String[] args)        {
 if (args.length > 0)            {
System.out.println("Command-line arguments:");

 for (int i = 0; i < args.length; i++)     {

 System.out.println("Argument " + (i + 1) + ": " + args[i]);

 }

 }
else {

System.out.println("No command-line arguments found.");    }

}

}
```

Output:

No command-line arguments found.

Practical 7: Write a program to implement different types of string methods.

```
package class_obj;
import class_obj.string_functions;
import java.util.*;
class string_functions    {
public static void main(String args[])     {
String a="HELLO";
String b="Java Programming";
System.out.println(a.toLowerCase());
System.out.println(b.toUpperCase());

System.out.println(a.concat(b));

System.out.println( "Length of the string is :"+a.length());

String c="Aachal";
System.out.println(c.trim());

System.out.println(c.charAt(2));

System.out.println(c.indexOf('l'));
System.out.println(b.equals(a));
```

```
        String d="Gita";


        System.out.println(d.replace('G','S'));
        System.out.println(d.isEmpty());  }     }
```

Output:

hello

JAVA PROGRAMMING

HELLOJava Programming

Length of the string is :5

Aachal

c

5

false

Sita

False

Practical 8: Write a program to implement function overloading.

```
package class_obj;
import class_obj.FunOverloading;
import java.util.*;
public class FunOverloading
{
public int add(int a, int b)
{
return a + b;
}
public double add(double a, double b,double c)      {
return a + b + c;     }
public static void main(String args[])      {
FunOverloading adder=new FunOverloading();
System.out.println("Sum of two integers:" + adder.add(20,10));
System.out.println("Sum of three integers:" + adder.add(20,10.7,50));
}
}
```

Output:

Sum of two integers:30
Sum of three integers:80.7

Prcatical 9: Write a program to demonstrate super keyword.

A.java class

```
package Superkey;

public class A {

	public int x;
	public void set()
	{
		x=10;
	}
}
```

B.java class

```
package Superkey;

public class B extends A
{
  int x;
	public void setvalue()
	{
		x=20;
	}
	public void display()
	{
		System.out.println(x+" "+super.x);
	}
}
```

Superset.java class

```
package Superkey;

public class superset {
	public static void main(String args[])
	{
	B b=new B();
	b.set();
	b.setvalue();
	b.display();

	}
}
```

Output:

20 10

Practical 10: Write the program to demonstrate a use of class and object.

```
package Program;
public class Java_practicals {
	String name;
	String sound;
	void makesound()
	{
		System.out.println(name + " makes the sound : "+sound);
	}
public static void main(String[] args)
{
	Java_practicals dog=new Java_practicals();
	dog.name="Dog";
	dog.sound="Bark";
	dog.makesound();
}
}
```

Output:

Dog makes the sound : Bark

Practical 11: Write the program to demonstrate student information using class and object.

```
package class_obj;

import class_obj.Student;

public class Student {

String name;

int age;

String grade;

//Constructor

Student(String name,int age,String grade)     {

this.name=name;

this.age=age;

this.grade=grade;    }

void display()    {

System.out.println("Student Name :"+ name);

System.out.println("Student Age :"+ age);
        System.out.println("Student Grade :"+ grade);    }

class Test    {

public static void main(String[] args)   {

Student s=new Student("ABC",12,"A");

s.display();   }  }   }
```

Output:

```
Student Name :ABC

Student Age :12
```

Practical 12: Write a program to demonstrate employee information using class and object.

```
package Program;

public class Employee   {

String name;

 int id;

 public Employee (String name, int id)        {

        this.name = name;

        this.id = id;        }

    public void displayInfo()    {

        System.out.println("Employee Information:");

        System.out.println("Name: " + name);

        System.out.println("ID: " + id);         }

          public static void main(String[] args)        {

          Employee employee = new Employee("ABC", 1001);

        employee.displayInfo();   }    }
```

Output:

Employee Information:

Name: ABC

ID: 1001

Practical 13: Write a program to implement interface in java.

```
package class_obj;
import class_obj.Animal;
public interface Animal {
void makesound();
}
class Dog implements Animal            {
public void makesound(){
System.out.println("Woof woofff");
}
}
class Cat implements Animal
{
public void makesound()           {
System.out.println("Meow!");   }
public class Test    {
public static void main(String args[])   {
        Animal dog=new Dog();
        Animal cat=new Cat();
        dog.makesound();
        cat.makesound();   }    }           }
```

Output:

```
Woof woofff
Meow!
```

Prcatical 14: Write a program to demonstrate use of local inner class.

```
package Program;
public class LocalInner{
        private int data=30;//instance variable
        void display(){
         class Local{
          void msg(){System.out.println(data);     }
         }
         Local l=new Local();
         l.msg();
        }
        public static void main(String args[]){
        LocalInner obj=new LocalInner();
         obj.display();
        }
       }
```

Output:

30

Practical 15: Write a program to demonstrate use of member inner class.

```
package Program;
public class Outerclass      {
private String message = "Hello from Outer Class!";
Class       {
        public void displayMessage()    {
            System.out.println("Message from Inner Class: " + message);
        }
    }
    public static void main(String[] args)       {
        Outerclass outer = new Outerclass();
        Outerclass.InnerClass inner = outer.new InnerClass();
        inner.displayMessage();
    }
}
```

Output:

Message from Inner Class: Hello from Outer Class**!**

Practical 16: Write program to demonstrate the built-in package.

```
package Program;
import java.util.*;
import java.math.*;
public class userdefinepackage      {
public static void main(String[] args)      {
        int a=10;
        int b=40;
        double c=10.40;
        System.out.println(Math.max(a,b));
        System.out.println(Math.min(a,b));
        System.out.println(Math.pow(a,b));
        System.out.println(Math.ceil(c));
        System.out.println(Math.floor(c));
}
}
```

Output:

```
40
10
1.0E40
11.0
10.0
```

Practical 17: Write a program to demonstrate the exception handling mechanism.

```
package Program;

import java.util.*;

public class Java_practicals     {

public static void main(String[] args)     {

int a,b,c;

System.out.println("Enter any two integers");

Scanner s=new Scanner(System.in);

a=s.nextInt();

b=s.nextInt();

try      {

c=a/b;

System.out.println("Divide is :" +c);       }

catch(ArithmeticException e)      {

System.out.println("You are trying to divide by zero");      }

finally    {

c=a*b;

System.out.println("Multiplication is :" +c);

System.out.println("Finally block always gets executed");

}

}

}
```

Output:

Enter any two integers

10

0

You are trying to divide by zero

Multiplication is :0

Finally block always gets executed

Practical 18: Write a program to demonstrate user defined exception.

```
package Program;
class AgeTooYoungException extends Exception     {
   public AgeTooYoungException(String message)       {
      super(message);
   }
}
public class AgeValidation       {
   public static void validateAge(int age) throws AgeTooYoungException {
      if (age < 18) {
         throw new AgeTooYoungException("Age must be 18 or older.");
      }
      System.out.println("Age is valid!");
   }
   public static void main(String[] args)
   {
    try     {
        int age = 16;
        System.out.println("Validating age: " + age);
         validateAge(age);
      }
        catch (AgeTooYoungException e)    {
         System.out.println("Exception caught: " + e.getMessage());
```

```
        } catch (Exception e) {
            System.out.println("An unexpected error occurred: " + e.getMessage());
        }
    }
}
```

Output :

Validating age: 16

Exception caught: Age must be 18 or older.

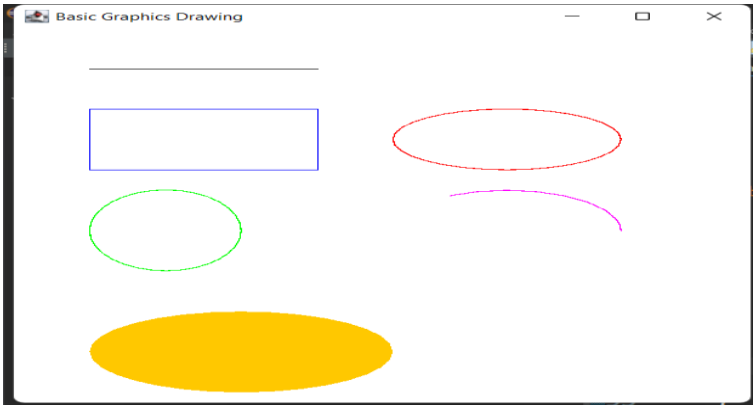
Practical 19 : Write a program to draw basic graphics construction like line,circle,arc,ellipse,rectangle and oval.

```
package Program;
import javax.swing.*;
import java.awt.*;
public class Basicgraph extends JPanel
{
	protected void paintComponent(Graphics g)
	{
		super.paintComponent(g);

		// Set a background color
		setBackground(Color.WHITE);

		// Draw a line
		g.setColor(Color.BLACK);
		g.drawLine(50, 50, 200, 50);

		// Draw a rectangle
		g.setColor(Color.BLUE);
		g.drawRect(50, 100, 150, 75);

		// Draw an oval (ellipse within a rectangle)
```

```
        g.setColor(Color.RED);
        g.drawOval(250, 100, 150, 75);

        // Draw a circle (oval with equal width and height)
        g.setColor(Color.GREEN);
        g.drawOval(50, 200, 100, 100);

        // Draw an arc
        g.setColor(Color.MAGENTA);
        g.drawArc(250, 200, 150, 100, 0, 120);

        // Draw an ellipse using fillOval for a solid effect
        g.setColor(Color.ORANGE);
        g.fillOval(50, 350, 200, 100);
    }
public static void main(String[] args) {

          // TODO Auto-generated method stub
           JFrame frame = new JFrame("Basic Graphics Drawing");
           Basicgraph graphicsPanel = new Basicgraph ();

        frame.add(graphicsPanel);
        frame.setSize(500, 500);
```

```
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setVisible(true);
        }
}
```

Output:



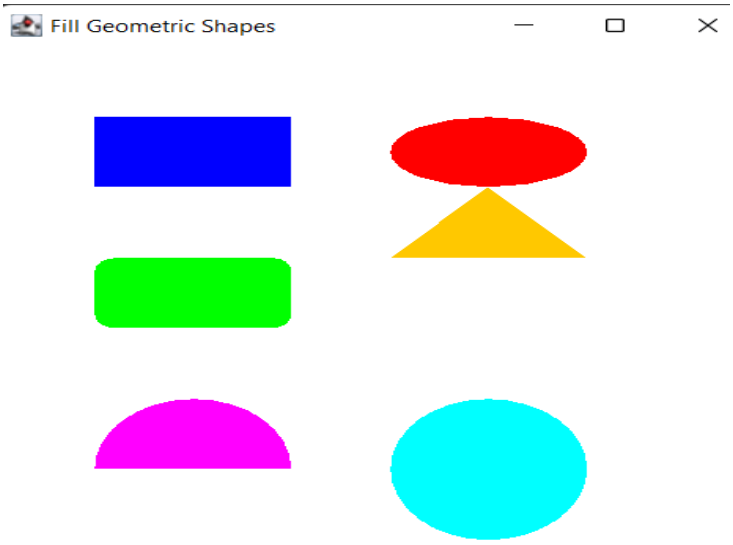
Practical 20: Write a program in java to draw fill geometric shape.

```
package Program;
import javax.swing.*;
import java.awt.*;
public class DrawFillShapes extends JPanel
{
        @Override
        protected void paintComponent(Graphics g) {
           super.paintComponent(g); // Clear the panel

           Graphics2D g2d = (Graphics2D) g;

           // Set background color
           setBackground(Color.WHITE);

           // Draw a filled rectangle
           g2d.setColor(Color.BLUE);
           g2d.fillRect(50, 50, 100, 50);

           // Draw a filled oval
           g2d.setColor(Color.RED);
           g2d.fillOval(200, 50, 100, 50);
```

```
        // Draw a filled rounded rectangle
        g2d.setColor(Color.GREEN);
        g2d.fillRoundRect(50, 150, 100, 50, 20, 20);

        // Draw a filled polygon (triangle)
        g2d.setColor(Color.ORANGE);
        int[] xPoints = {200, 250, 300};
        int[] yPoints = {150, 100, 150};
        g2d.fillPolygon(xPoints, yPoints, 3);

        // Draw a filled arc
        g2d.setColor(Color.MAGENTA);
        g2d.fillArc(50, 250, 100, 100, 0, 180);

        // Draw a filled circle
        g2d.setColor(Color.CYAN);
        g2d.fillOval(200, 250, 100, 100);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Fill Geometric Shapes");
        DrawFillShapes panel = new DrawFillShapes();
```

```
        frame.add(panel);
        frame.setSize(400, 400);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

Output:



Practical 21: Write a program in java to display messges in various fonts in a frame.

```
package Program;

import javax.swing.*;

import java.awt.*;

public class FontDisplay  extends JPanel

{

	protected void paintComponent(Graphics g) {

		super.paintComponent(g);


		// Cast to Graphics2D for better font control

		Graphics2D g2d = (Graphics2D) g;


		// Set the background color

		setBackground(Color.WHITE);


		// Example message

		String message = "Hello, World!";


		// Font 1: Serif, Plain

		g2d.setFont(new Font("Serif", Font.PLAIN, 20));

		g2d.setColor(Color.BLUE);

		g2d.drawString(message, 50, 50);
```

```
        // Font 2: SansSerif, Bold
        g2d.setFont(new Font("SansSerif", Font.BOLD, 24));
        g2d.setColor(Color.RED);
        g2d.drawString(message, 50, 100);



        g2d.setFont(new Font("Monospaced", Font.ITALIC, 28));
        g2d.setColor(Color.GREEN);
        g2d.drawString(message, 50, 150);

        // Font 4: Dialog, Bold + Italic
        g2d.setFont(new Font("Dialog", Font.BOLD | Font.ITALIC, 32));
        g2d.setColor(Color.MAGENTA);
        g2d.drawString(message, 50, 200);

        // Font 5: DialogInput, Plain
        g2d.setFont(new Font("DialogInput", Font.PLAIN, 36));
        g2d.setColor(Color.ORANGE);
        g2d.drawString(message, 50, 250);
    }
  public static void main(String[] args) {
          JFrame frame = new JFrame("Font Display Example");
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setSize(500, 400);


        // Add custom panel
        FontDisplay fontPanel = new FontDisplay ();
        frame.add(fontPanel);
        frame.setVisible(true); // Display the frame
          }
}
```

Output:

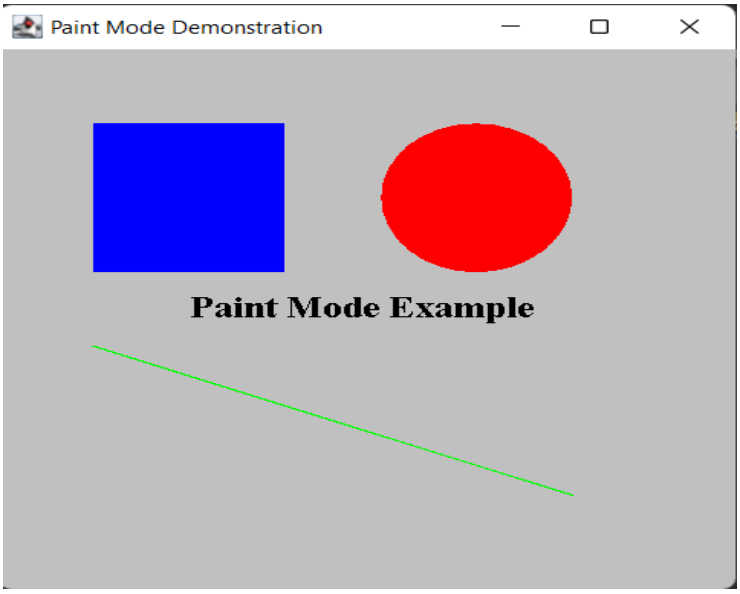
Practical 22: Write a program in java to demonstrate paint mode.

```
package Program;
import javax.swing.*;
import java.awt.*;
public class paintmode extends JFrame {
    // Constructor to set up the JFrame
    public Java_practicals() {
        setTitle("Paint Mode Demonstration");
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        add(new CustomPanel()); // Add a custom JPanel
    }

    // Custom JPanel for painting
    class CustomPanel extends JPanel {
        @Override
        protected void paintComponent(Graphics g) {
            super.paintComponent(g); // Clear the panel

            // Enable 2D graphics for more control
            Graphics2D g2d = (Graphics2D) g;

            // Set background color (to be visible behind objects)
```

```
        setBackground(Color.LIGHT_GRAY);

        // Draw a filled rectangle
        g2d.setPaint(Color.BLUE);
        g2d.fillRect(50, 50, 100, 100);

        // Draw a filled ellipse
        g2d.setPaint(Color.RED);
        g2d.fillOval(200, 50, 100, 100);

        // Draw a diagonal line
        g2d.setPaint(Color.GREEN);
        g2d.drawLine(50, 200, 300, 300);

        // Draw text
        g2d.setPaint(Color.BLACK);
        g2d.setFont(new Font("Serif", Font.BOLD, 20));
        g2d.drawString("Paint Mode Example", 100, 180);
    }
}

// Main method to run the program
public static void main(String[] args) {
```

```
        SwingUtilities.invokeLater(() -> {
          paintmode demo = new paintmode();
            demo.setVisible(true);
        });
    }
}
```

Output:



Practicals 23: Write a program to draw a smiley face.

```
package Program;
import javax.swing.*;
import java.awt.*;
public class SmileyFace extends JPanel {

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Cast to Graphics2D for better rendering
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);

        // Draw face (circle)
        g2d.setColor(Color.YELLOW);
        g2d.fillOval(50, 50, 200, 200); // x, y, width, height

        // Draw eyes
        g2d.setColor(Color.BLACK);
        g2d.fillOval(100, 100, 20, 20); // Left eye
        g2d.fillOval(180, 100, 20, 20); // Right eye
```

```
        // Draw smile
        g2d.setColor(Color.BLACK);
        g2d.drawArc(90, 120, 120, 80, 0, -180); // x, y, width, height, startAngle, arcAngle
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Smiley Face");
        SmileyFace smiley = new SmileyFace();

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 300);
        frame.add(smiley);
        frame.setVisible(true);
    }
}
```

Output:

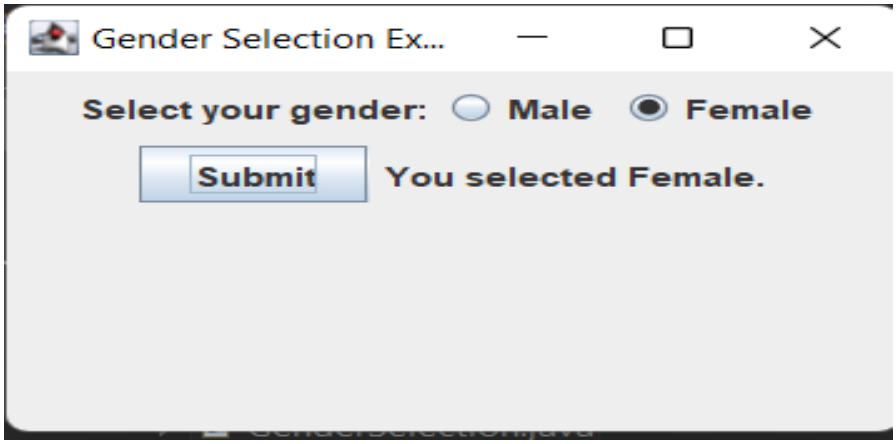
Practical 24: Write a program in java to demonstrate user interface component radio button.

```
package Program;

import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class GenderSelection {

    public static void main(String[] args) {

        // Create a JFrame

        JFrame frame = new JFrame("Gender Selection Example");

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setSize(300, 200);

        frame.setLayout(new FlowLayout());


        // Create a label

        JLabel label = new JLabel("Select your gender:");


        // Create radio buttons

        JRadioButton maleButton = new JRadioButton("Male");

        JRadioButton femaleButton = new JRadioButton("Female");


        // Add radio buttons to a ButtonGroup
```

```
ButtonGroup group = new ButtonGroup();
group.add(maleButton);
group.add(femaleButton);

// Create a submit button
JButton submitButton = new JButton("Submit");
JLabel resultLabel = new JLabel();

// Add ActionListener to the submit button
submitButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (maleButton.isSelected()) {
            resultLabel.setText("You selected Male.");
        } else if (femaleButton.isSelected()) {
            resultLabel.setText("You selected Female.");
        } else {
            resultLabel.setText("No gender selected!");
        }
    }
});

// Add components to the frame
```

```
        frame.add(label);
        frame.add(maleButton);
        frame.add(femaleButton);
        frame.add(submitButton);
        frame.add(resultLabel);
        // Set frame visibility
        frame.setVisible(true);
    }
}
```

Output:



Practical 25: Write a program in java JButton using swing.

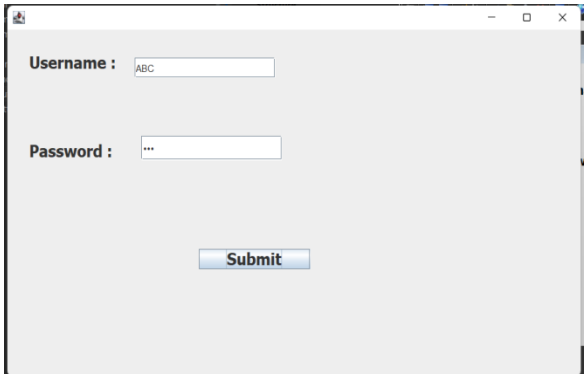
```
package Program;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class J_button extends JFrame {
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;
	private JTextField textField;
	private JPasswordField passwordField;
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					J_button frame = new J_button();
```

```
                        frame.setVisible(true);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            });
    }
    public J_button() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 771, 464);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Username :");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));
        lblNewLabel.setBounds(29, 28, 125, 25);
        contentPane.add(lblNewLabel);

        textField = new JTextField();
        textField.setBounds(168, 35, 186, 25);
```

```
		contentPane.add(textField);
		textField.setColumns(10);

		passwordField = new JPasswordField();
		passwordField.setBounds(177, 131, 186, 30);
		contentPane.add(passwordField);

		JLabel lblNewLabel_1 = new JLabel("Password : ");
		lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 20));
		lblNewLabel_1.setBounds(29, 134, 125, 30);
		contentPane.add(lblNewLabel_1);

		JButton btnNewButton = new JButton("Submit");
		btnNewButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				System.out.println("Login Successfully");
				System.out.println("User name is : "
+textField.getText());
				System.out.println("Password is : " +
passwordField.getText());
			}
		});
		btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 20));
```

```
        btnNewButton.setBounds(253, 270, 147, 25);
        contentPane.add(btnNewButton);
    } }
```

Output :



Login Successfully

User name is : ABC

Password is : 123

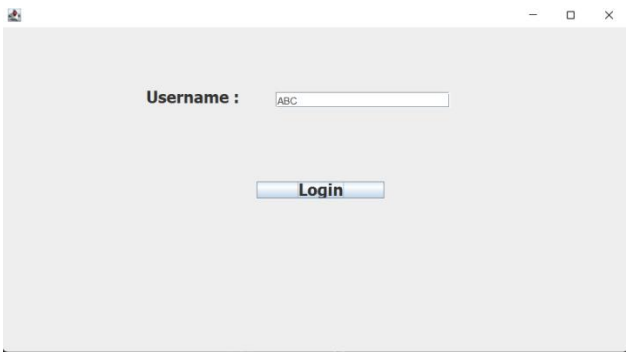
Practical 26: Write a program in java to demonstrate java JTextField using swing.

```
package Program;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class Jtexfield extends JFrame	{
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;
	private JTextField textField;
	public static void main(String[] args)	{
		EventQueue.invokeLater(new Runnable()	{
			public void run()	{
				try {
					Jtexfield frame = new Jtexfield();
					frame.setVisible(true);
```

```
                } catch (Exception e) {
                        e.printStackTrace();
                }
            }
        });
}
public Jtexfield() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 786, 435);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Username :");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));
        lblNewLabel.setBounds(178, 73, 116, 25);
        contentPane.add(lblNewLabel);

        textField = new JTextField();
        textField.setBounds(337, 80, 215, 19);
        contentPane.add(textField);
        textField.setColumns(10);
```

```
        JButton btnNewButton = new JButton("Login");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

            System.out.println("Login Successfully");
            System.out.println("User name is : " +textField.getText());
            }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 20));
        btnNewButton.setBounds(314, 189, 157, 21);
        contentPane.add(btnNewButton);
    }
}
```

Output :



Login Successfully

User name is : ABC

Practical 27: Write a program to demonstrate Java JTextArea using swing.

```
package Program;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JTextArea;
public class Jtexfield extends JFrame {
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Jtexfield frame = new Jtexfield();
					frame.setVisible(true);
				} catch (Exception e) {
```

```
                    e.printStackTrace();
                }
            }
        });
}
public Jtexfield() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 786, 435);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Introduce Yourself");
        lblNewLabel.setBounds(206, 34, 300, 44);
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 30));
        contentPane.add(lblNewLabel);

        JTextArea textArea = new JTextArea();
        textArea.setBounds(172, 88, 404, 284);
        contentPane.add(textArea);
```

```
        JLabel lblNewLabel_1 = new JLabel("New label");
        lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 27));
        lblNewLabel_1.setBounds(57, 189, 45, 13);
        contentPane.add(lblNewLabel_1);
    }
}
```

Output :

Practical 28: Write a program to demonstrate Java JPasswordField using swing.

```
package Program;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JLabel;

import java.awt.Font;

import javax.swing.JTextField;

import javax.swing.JPasswordField;

import javax.swing.JCheckBox;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Jpassword extends JFrame {

	private static final long serialVersionUID = 1L;

	private JPanel contentPane;

	private JTextField textField;

	private JPasswordField passwordField;

	public static void main(String[] args) {

		EventQueue.invokeLater(new Runnable() {

			public void run() {
```

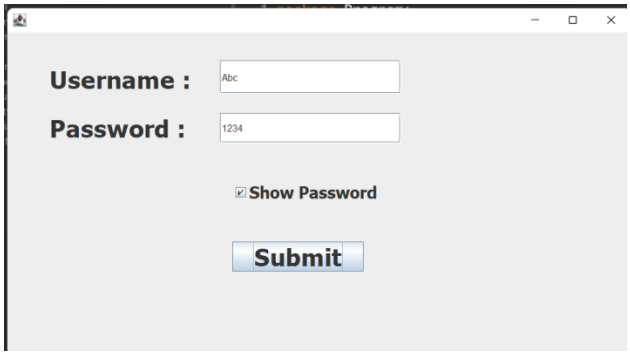
```
                    try {
                        Jpassword frame = new Jpassword();
                        frame.setVisible(true);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            });
}
public Jpassword() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 782, 435);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Username : ");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 30));
        lblNewLabel.setBounds(52, 35, 200, 47);
        contentPane.add(lblNewLabel);
```

```
		textField = new JTextField();
		textField.setBounds(262, 35, 223, 41);
		contentPane.add(textField);
		textField.setColumns(10);

		JLabel lblNewLabel_1 = new JLabel("Password :");
		lblNewLabel_1.setFont(new Font("Tahoma", Font.BOLD, 30));
		lblNewLabel_1.setBounds(52, 100, 168, 37);
		contentPane.add(lblNewLabel_1);

		passwordField = new JPasswordField();
		passwordField.setBounds(262, 100, 223, 37);
		contentPane.add(passwordField);

		JCheckBox chckbxNewCheckBox = new JCheckBox("Show
Password");
		chckbxNewCheckBox.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				if(chckbxNewCheckBox.isSelected())
				{
					passwordField.setEchoChar((char) 0);
				}
				else
```

```
                {
                        passwordField.setEchoChar('*');
                }
        }
    });

    chckbxNewCheckBox.setFont(new Font("Tahoma", Font.BOLD,
20));
    chckbxNewCheckBox.setBounds(277, 187, 208, 21);
    contentPane.add(chckbxNewCheckBox);

    JButton btnNewButton = new JButton("Submit");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            System.out.println("Login Successfully");
            System.out.println("User name is : "
+textField.getText());
            System.out.println("Password is : "
+passwordField.getText());

        }
    });
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 30));
    btnNewButton.setBounds(277, 258, 162, 37);
```

```
            contentPane.add(btnNewButton);
        }
}
```

Output :



Login Successfully

User name is : Abc

Password is : 1234

Practical 29: Write a program to demonstrate Java JCheckbox using swing.

```
package Program;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JLabel;
import java.awt.Font;
import javax.swing.JCheckBox;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
public class Jcheckbox extends JFrame {
	private static final long serialVersionUID = 1L;
	private JPanel contentPane;
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					Jcheckbox frame = new Jcheckbox();
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
```

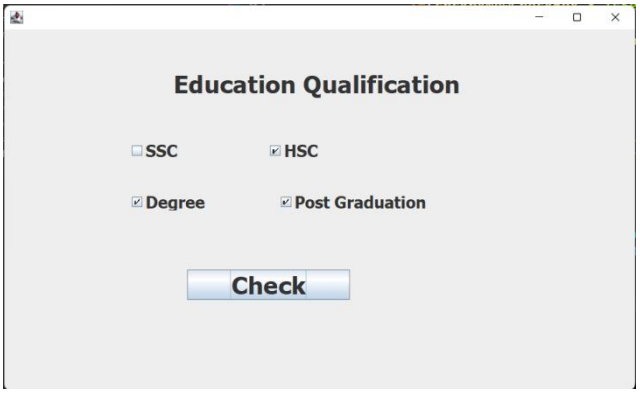
```
                }
            }
        });
    }
    public Jcheckbox() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 790, 481);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Education Qualification");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 30));
        lblNewLabel.setBounds(208, 49, 352, 37);
        contentPane.add(lblNewLabel);

        JCheckBox chckbxNewCheckBox = new JCheckBox("SSC");
        chckbxNewCheckBox.setFont(new Font("Tahoma", Font.BOLD,
20));
        chckbxNewCheckBox.setBounds(153, 139, 140, 21);
        contentPane.add(chckbxNewCheckBox);
```

```
		JCheckBox chckbxNewCheckBox_1 = new JCheckBox("HSC");
		JCheckBox ssc = new JCheckBox();
		chckbxNewCheckBox_1.setFont(new Font("Tahoma", Font.BOLD,
20));
		chckbxNewCheckBox_1.setBounds(323, 139, 93, 21);
		contentPane.add(chckbxNewCheckBox_1);

		JCheckBox chckbxNewCheckBox_2 = new JCheckBox("Degree");
		JCheckBox  hsc= new JCheckBox();
		chckbxNewCheckBox_2.setFont(new Font("Tahoma", Font.BOLD,
20));
		chckbxNewCheckBox_2.setBounds(153, 202, 112, 21);
		contentPane.add(chckbxNewCheckBox_2);

		JCheckBox chckbxNewCheckBox_3 = new JCheckBox("Post
Graduation");
		JCheckBox pg = new JCheckBox();
		chckbxNewCheckBox_3.setFont(new Font("Tahoma", Font.BOLD,
20));
		chckbxNewCheckBox_3.setBounds(336, 202, 187, 21);
		contentPane.add(chckbxNewCheckBox_3);

		JButton btnNewButton = new JButton("Check");
		btnNewButton.addActionListener(new ActionListener() {
```

```
public void actionPerformed(ActionEvent e) {
        String qul[]=new String[5];



        if(chckbxNewCheckBox_1.isSelected())
        {
                qul[0]=chckbxNewCheckBox_1.getText();
        }
        if(chckbxNewCheckBox_2.isSelected())
        {
                qul[1]=chckbxNewCheckBox_2.getText();


        }


        if(chckbxNewCheckBox_3.isSelected())
        {
                qul[3]=chckbxNewCheckBox_3.getText();
        }
        for(int i=0;i<=3;i++)
        {
                System.out.println(qul[i]);
        }
}
```

```
        });
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 30));
        btnNewButton.setBounds(225, 296, 200, 37);
        contentPane.add(btnNewButton);
    }
}
```

Output :



HSC

Degree

null

Post Graduation

Practical 30: Write the program to demonstrate Java JComboBox using swing.

```
package Program;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.border.EmptyBorder;

import javax.swing.JComboBox;

import javax.swing.DefaultComboBoxModel;

import java.awt.Font;

import javax.swing.JLabel;

import javax.swing.JButton;

import java.awt.event.ActionListener;

import java.awt.event.ActionEvent;

public class Jcombobo extends JFrame {


	private static final long serialVersionUID = 1L;

	private JPanel contentPane;

	public static void main(String[] args) {

		EventQueue.invokeLater(new Runnable() {

			public void run() {

				try {

					Jcombobo frame = new Jcombobo();
```

```
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
    public Jcombobo() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 698, 445);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JComboBox comboBox = new JComboBox();
        comboBox.setFont(new Font("Tahoma", Font.BOLD, 30));
        comboBox.setModel(new DefaultComboBoxModel(new String[]
{"Select City", "Pune", "Mumbai", "Nashik", "Dhule", "Aurangabad"}));
        comboBox.setBounds(106, 10, 425, 48);
        contentPane.add(comboBox);
```

```
        JLabel lblNewLabel = new JLabel("");
        lblNewLabel.setFont(new Font("Tahoma", Font.BOLD, 20));
        lblNewLabel.setBounds(511, 332, 163, 34);
        contentPane.add(lblNewLabel);


        JButton btnNewButton = new JButton("Ok");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String label=(String) (comboBox.getSelectedItem());
                lblNewLabel.setText(label);
            }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.BOLD, 21));
        btnNewButton.setBounds(268, 182, 85, 21);
        contentPane.add(btnNewButton);
    }
}
```

Output :