JAVA PRACTICALS

1.TABLE(prac no 1)

```
import java.awt.*;

import java.awt.event.*;


public class table extends Frame implements ActionListener {

    Label l1, l2;

    TextField t1;

    TextArea t2;

    Button b1;


    table() {

        l1 = new Label("Enter a number");

        l1.setBounds(10, 30, 80, 70);

        add(l1);
        t1 = new TextField();
        t1.setBounds(100, 30, 80, 20);
        add(t1);

        l2 = new Label("Table is:");
        l2.setBounds(10, 90, 90, 50);
        add(l2);

        t2 = new TextArea();
        t2.setBounds(100, 90, 80, 100);
        add(t2);

        b1 = new Button("Show Table");
        b1.setBounds(10, 200, 90, 30); // Adjusted button
bounds
        add(b1);
        b1.addActionListener(this);

        addWindowListener(new WindowAdapter() {
```

```
        public void windowClosing(WindowEvent e) {

            dispose();

        }

    });


    setSize(300, 300);

    setLayout(null);

    setVisible(true);

}


public static void main(String args[]) {

    table t = new table();

}


public void actionPerformed(ActionEvent e) {

    String s1 = t1.getText();

    int a = Integer.parseInt(s1);

    if (e.getSource() == b1) {
            String result = "";
            for (int x = 1; x <= 10; x++) {
                result += x * a + "\n";
            }
            t2.setText(result);
        }
    }
}
```

2. area of triangle(prac no 3)

```
import java.util.*;
public class area{
public static void main(String args[]){
area a=new area();
double z;
Scanner s=new Scanner(System.in);
System.out.println("Enter triangle's base:");
int x=s.nextInt();
System.out.println("Enter a triangle's height:");
int y=s.nextInt();
z=0.5*x*y;
System.out.println("Area of Triangle is: \t" + (z));
}
}
```

3. FACTORIAL(prac no 4)

```
import java.awt.*;
import java.awt.event.*;

public class factorial extends Frame implements ActionListener {
        Label l1,l2;
        TextField t1,t2;
        Button b1;
```

```
factorial() {
        l1 = new Label("Enter a number:");
        l1.setBounds(10, 30, 100, 20);
        add(l1);

        l2=new Label("factorial is:");
        l2.setBounds(10,60,100,20);
        add(l2);

        t1 = new TextField();
        t1.setBounds(120, 30, 80, 20);
        add(t1);

        t2=new TextField();
        t2.setBounds(120,60,80,20);
        add(t2);

        b1 = new Button("Calculate Factorial");
        b1.setBounds(10, 90, 150, 30);
        add(b1);
        b1.addActionListener(this);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        setSize(300, 150);
        setLayout(null);
        setVisible(true);
    }

    public static void main(String args[]) {
        factorial f = new factorial();
    }
    public void actionPerformed(ActionEvent e) {
        String s1 = t1.getText();
        int a = Integer.parseInt(s1);
        int sum = 1;
        if (e.getSource() == b1) {
            for (int x = 1; x <= a; x++) {
                sum *= x;
            }
            t2.setText(Integer.toString(sum));
        }
    }
}
```

4. EVEN OR NOT(prac no 6)

```
import java.awt.*;
import java.awt.event.*;
```

```
public class factorial extends Frame implements
ActionListener {
        Label l1,l2;
        TextField t1,t2;
        Button b1;

        factorial() {
                l1 = new Label("Enter a number:");
                l1.setBounds(10, 30, 100, 20);
                add(l1);

                l2=new Label("Even/Odd");
                l2.setBounds(10,60,100,20);
                add(l2);

                t1 = new TextField();
                t1.setBounds(120, 30, 80, 20);
                add(t1);
```

```
t2=new TextField();
t2.setBounds(120,60,100,20);
add(t2);

b1 = new Button("Check");
b1.setBounds(10, 90, 150, 30);
add(b1);
b1.addActionListener(this);

addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
                dispose();
        }
});

setSize(300, 150);
setLayout(null);
```

```
        setVisible(true);
    }

    public static void main(String args[]) {
        factorial f = new factorial();
    }

    public void actionPerformed(ActionEvent e) {
        String s1 = t1.getText();
        int a = Integer.parseInt(s1);
        if(e.getSource()==b1){
            if(a%2==0){
                t2.setText("Number is even");
            }
            else{
                t2.setText("Number is odd");
            }
        }
    }
}
```

5. REVERSE(prac no 8)

```
import java.awt.*;
import java.awt.event.*;

public class factorial extends Frame implements ActionListener {
    Label l1,l2;
    TextField t1,t2;
    Button b1;

    factorial() {
        l1 = new Label("Enter a number:");
        l1.setBounds(10, 30, 100, 20);
        add(l1);
```

```
l2=new Label("Reverse is:");
l2.setBounds(10,60,100,20);
add(l2);

t1 = new TextField();
t1.setBounds(120, 30, 80, 20);
add(t1);

t2=new TextField();
t2.setBounds(120,60,100,20);
add(t2);

b1 = new Button("Reverse");
b1.setBounds(10, 90, 150, 30);
add(b1);
b1.addActionListener(this);
```

```
           addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            dispose();
        }
    });



            setSize(300, 150);
            setLayout(null);
            setVisible(true);
        }

        public static void main(String args[]) {
            factorial f = new factorial();
        }

        public void actionPerformed(ActionEvent e) {
            String s1 = t1.getText();
            int a = Integer.parseInt(s1);

            int n=a;

            int last;

            int reverse=0;

            if(e.getSource()==b1){

                while(n!=0){

                    last = n%10;

                    n= n/10;

                    reverse=reverse*10+last;

                }

                t2.setText(Integer.toString(reverse));

            }

        }

    }
```

6. PALLINDROME(prac no 13)

```
import java.awt.*;
import java.awt.event.*;


public class factorial extends Frame implements
ActionListener {
        Label l1,l2;
        TextField t1,t2;
        Button b1;

        factorial() {
                l1 = new Label("Enter a number:");
                l1.setBounds(10, 30, 100, 20);
                add(l1);

                l2=new Label("check for pallindrome");
                l2.setBounds(10,60,150,20);
                add(l2);

                t1 = new TextField();
                t1.setBounds(120, 30, 80, 20);
                add(t1);


                t2=new TextField();
                t2.setBounds(120,60,200,20);
                add(t2);


                b1 = new Button("check");
                b1.setBounds(10, 90, 150, 30);
                add(b1);
                b1.addActionListener(this);


               addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
                dispose();
        }
});
```

```
    setSize(300, 150);
    setLayout(null);
    setVisible(true);
}

public static void main(String args[]) {
    factorial f = new factorial();
}

public void actionPerformed(ActionEvent e) {
    String s1 = t1.getText();
    int a = Integer.parseInt(s1);
    int n=a;
    int last;
    int reverse=0;
    if(e.getSource()==b1){
        while(n!=0){
            last = n%10;
```

```
                n= n/10;
                reverse=reverse*10+last;
            }
            if(n==a){
                t2.setText("Number is Pallindome");
            }
            else{
                t2.setText("number is not pallindrome");
            }
        }
    }
}
```

7. INTERFACE(applicable for 35 and 7)

```
interface area{
void areaCompute(float x, float y);
}
```

```
class Rectangle implements area{
public void areaCompute(float x, float y){
System.out.println("Area of Rectangle is: \t" + (x*y));
}
}
class Circle implements area{
public void areaCompute(float x, float y){
System.out.println("Area of triangle is: \t" + (0.5*x*y));
}
}
class main{
public static void main(String args[]){
Rectangle r=new Rectangle();
Circle c=new Circle();
r.areaCompute(4,8);
c.areaCompute(9,7);
}
}
```

8. USER DEFINED EXCEPTION(prac no 31 and can be used for 14)

it was executing only on online compiler

```
public class myException extends Exception{
        public myException(String st){
                super(st);
        }
}
// can be executed on your system..

class Student {
// when executing online mention class with public modifier
         void check(int roll, String name, int age, String
course) throws myException {
            if (age < 15 || age > 21) {
                throw new myException("Age not
within the range");
            } else {
                System.out.println("Valid");
            }
    }
     public static void main(String args[]) {
         Student s = new Student();
        try{
         s.check(14, "kanha", 22, "bms");
     }catch(myException e){
        System.out.println("Error:\t" +
(e.getMessage()));
}
    }
    }
```

9. USER-DEFINED PACKAGES(prac no 29 and 9)

```
//first create a main folder named "xyz"

//Inside it create a folder named "prime",below is the code for it

package prime;

public class radha{

public int add(int a, int b){

return a+b;

}

public int subtract(int a, int b){

return a-b;

}

}


// inside main folder create one more folder named
```

"factorial",below is the code for it

```
package factorial;

public class kanha{

public int mul(int a, int b){

return a*b;

}

public int div(int a, int b){

return a/b;

}

}

//Inside main folder write a java program

import prime.radha;

import factorial.kanha;

public class primeFact{

        public static void main(String args[]){

radha r=new radha();

kanha k=new kanha();
int result1 = r.add(5,9);
System.out.println("Addition is: \t" + (result1));
int result2 = r.subtract(5,9);
System.out.println("Subtraction is:\t" + (result2));
int result3 = k.mul(5,9);
System.out.println("Addition is: \t" + (result3));
int result4 = k.div(5,9);
System.out.println("Addition is: \t" + (result4));
}
}
// Execute this one code but also compile rest 3
```

10. ARITHMETIC OPERATORS(prac no 18)

```
import java.awt.*;
import java.awt.event.*;

public class factorial extends Frame implements
```

```
ActionListener {
    Label l1, l2, l3;
    TextField t1, t2, t3;
    Button b1, b2, b3, b4, b5;

    factorial() {
        l1 = new Label("Enter first number:");
        l1.setBounds(10, 30, 100, 20);
        add(l1);

        l2 = new Label("Enter second number:");
        l2.setBounds(10, 60, 150, 20);
        add(l2);

        l3 = new Label("Result:");
        l3.setBounds(10, 90, 150, 20);
        add(l3);
```

```
t1 = new TextField();

t1.setBounds(150, 30, 120, 20);

add(t1);


t2 = new TextField();

t2.setBounds(150, 60, 120, 20);

add(t2);


t3 = new TextField();

t3.setBounds(150, 90, 120, 20);

add(t3);


b1 = new Button("Add");

b1.setBounds(10, 120, 70, 30);

add(b1);

b1.addActionListener(this);
```

```
b2 = new Button("Subtract");
b2.setBounds(90, 120, 70, 30);
add(b2);
b2.addActionListener(this);

b3 = new Button("Multiply");
b3.setBounds(170, 120, 70, 30);
add(b3);
b3.addActionListener(this);

b4 = new Button("Divide");
b4.setBounds(250, 120, 70, 30);
add(b4);
b4.addActionListener(this);

b5 = new Button("Modulus");
```

```
        b5.setBounds(330, 120, 70, 30);
        add(b5);
        b5.addActionListener(this);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });

        setSize(420, 180);
        setLayout(null);
        setVisible(true);
    }

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

```
    factorial f = new factorial();
}

public void actionPerformed(ActionEvent e) {
    String s1 = t1.getText();
    String s2 = t2.getText();
    int a = Integer.parseInt(s1);
    int b = Integer.parseInt(s2);
    int c = 0;

    if (e.getSource() == b1) {
        c = a + b;
    } else if (e.getSource() == b2) {
        c = a - b;
    } else if (e.getSource() == b3) {
        c = a * b;
    } else if (e.getSource() == b4) {
            if (b != 0) {
                c = a / b;
            } else {
                t3.setText("Cannot divide by zero!");
                return;
            }
        } else if (e.getSource() == b5) {
            if (b != 0) {
                c = a % b;
            } else {
                t3.setText("Cannot mod by zero!");
                return;
            }
        }
        t3.setText(Integer.toString(c));
    }
}
```

11. MOUSE LISTENER AND MOUSE MOTION EVENTS

```
import java.awt.*;

import java.awt.event.*;

public class mouse extends Frame implements
MouseListener,MouseMotionListener{

Label l;

mouse(){

addMouseListener(this);

addMouseMotionListener(this);

l=new Label();

l.setBounds(20,40,80,20);

add(l);

setSize(400,300);

setLayout(null);

setVisible(true);

addWindowListener(new WindowAdapter(){
```

```
public void windowClosing(WindowEvent e){
        dispose();
}
});
}
public void mouseEntered(MouseEvent e){
l.setText("Mouse Entered");
}
public void mouseExited(MouseEvent e){
l.setText("Mouse Exited");
}

public void mousePressed(MouseEvent e){
l.setText("Mouse Pressed");
}
public void mouseReleased(MouseEvent e){
l.setText("Mouse Released");
```

```
}
public void mouseClicked(MouseEvent e){
l.setText("Mouse Clicked");
}
public void mouseDragged(MouseEvent e){
l.setText("Mouse Dragged");
}
public void mouseMoved(MouseEvent e){
l.setText("Mouse Moved");
}
public static void main(String args[]){
mouse m=new mouse();
}
}
```

12. PRIME OR NOT(prac no 12)

```
import java.awt.*;
import java.awt.event.*;
```

```
public class     prime extends Frame implements
ActionListener{

Label l1;

TextField t1;

Button b1;

prime(){


l1=new Label("Enter a number:");

l1.setBounds(20, 40,100,20);

add(l1);


t1=new TextField();

t1.setBounds(120,40,90,20);

add(t1);


b1=new Button("click me");

b1.setBounds(100,90,80,20);
add(b1);
b1.addActionListener(this);

addWindowListener(new WindowAdapter(){
public void windowClosing(WindowEvent e){
        dispose();
}
});

setSize(400,300);
setLayout(null);
setVisible(true);
}

public static void main(String args[]){
prime p=new prime();
}
```

```
public void actionPerformed(ActionEvent e){
String s1=t1.getText();
int a = Integer.parseInt(s1);
int y = 0;
if(e.getSource()==b1){

for(int x=2;x<=a;x++)
{
if(a%x==0){
y++;
}
if(y<=1){
t1.setText("Prime");
}
else{
t1.setText("Not Prime");
}

}

}

}

}
```

13. FOR PRAC 2

```
import java.util.Scanner;


public class BankAccount {

    // Data Members

    private String depositorName;

    private long accountNumber;

    private String accountType;

    private double balance;


    // Method to read account details

    public void readAccountDetails() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter depositor's name: ");

        depositorName = scanner.nextLine();

        System.out.print("Enter account number: ");

        accountNumber = scanner.nextLong();

        scanner.nextLine(); // Consume newline

        System.out.print("Enter account type
(Savings/Current): ");

        accountType = scanner.nextLine();

        balance = 500.00; // Initialize balance with
minimum balance

        System.out.println("Account created
successfully!");

    }


    // Method to deposit amount

    public void deposit(double amount) {

        balance += amount;
        System.out.println("Deposit successful.");
    }

    // Method to withdraw amount
    public void withdraw(double amount) {
        if (balance - amount >= 500.00) {
            balance -= amount;
            System.out.println("Withdrawal successful.");
        } else {
            System.out.println("Insufficient balance! Minimum balance must be maintained.");
        }
    }

    // Method to display balance
    public void displayBalance() {
```

```
    System.out.println("Balance: Rs." + balance);
}

// Main method to test the BankAccount class
public static void main(String[] args) {
    BankAccount account = new BankAccount();
    account.readAccountDetails();

    // Deposit some amount
    account.deposit(1000.00);

    // Withdraw some amount
    account.withdraw(300.00);

    // Display balance
    account.displayBalance();
}
```

```
}
```

14. PRAC NO 5

```
import java.util.Scanner;

// Employee class
class Employee {
        // Data members
        String name;
        double salary;

        // Constructor
        public Employee(String name, double salary) {
                this.name = name;
                this.salary = salary;
        }
```

```
    // Method to read employee information
    public void readInfo() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter employee name: ");
        name = scanner.nextLine();
        System.out.print("Enter employee salary: ");
        salary = scanner.nextDouble();
    }

    // Method to display employee information
    public void displayInfo() {
        System.out.println("Employee name: " + name);
        System.out.println("Employee salary: " + salary);
    }
}

// Manager class (inherits from Employee)
class Manager extends Employee {
    // Data member
    String department;

    // Constructor
    public Manager(String name, double salary, String
department) {
        super(name, salary);
        this.department = department;
    }

    // Method to read manager information
    public void readInfo() {
        super.readInfo(); // Call parent class method
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter manager department: ");
        department = scanner.nextLine();
    }

    // Method to display manager information
    public void displayInfo() {
        super.displayInfo(); // Call parent class method
        System.out.println("Manager department: " + department);
    }
}

// Executive class (inherits from Manager)
class Executive extends Manager {
    // Data member
    String location;

    // Constructor
    public Executive(String name, double salary, String department, String location) {
        super(name, salary, department);
        this.location = location;
    }
    // Method to read executive information
    public void readInfo() {
        super.readInfo(); // Call parent class method
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter executive location: ");
        location = scanner.nextLine();
    }

    // Method to display executive information
    public void displayInfo() {
        super.displayInfo(); // Call parent class method
        System.out.println("Executive location: " + location);
    }
}

// Main class to test Employee, Manager, and Executive classes
public class main {
    public static void main(String[] args) {
        Executive executive = new Executive("John", 10000, "Management", "New York");
        executive.displayInfo();
    }
}
```

15. PRAC NO. 15

```
public class Player {
    // Data members
    String pName;
    int innings;
    int runs;
    int notOuts;

    // Method to display player details
    public void showData() {
        System.out.println("Player Name: " + pName);
        System.out.println("Innings: " + innings);
        System.out.println("Runs: " + runs);
        System.out.println("Not Outs: " + notOuts);
    }

    // Method to calculate batting average
    public void calcAvg() {
        // Calculate batting average
        double battingAvg = runs / (double) (innings -
notOuts);
        System.out.println("Batting Average: " +
battingAvg);
    }

    // Main method to test Player class
    public static void main(String[] args) {
        // Create a player object
        Player player1 = new Player();
        // Set player details
        player1.pName = "John";
        player1.innings = 20;
        player1.runs = 500;
        player1.notOuts = 5;


        // Display player details
        player1.showData();
        // Calculate and display batting average
        player1.calcAvg();
    }
}
```

16.PRAC NO 11

```
import java.util.Scanner;


public class SalaryCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);


        // Input basic salary
        System.out.print("Enter basic salary (Rs.): ");
        double basicSalary = scanner.nextDouble();
        // Calculate net salary based on criteria
        double netSalary = calculateNetSalary(basicSalary);

        // Display net salary
        System.out.println("Net Salary (Rs.): " + netSalary);

        scanner.close();
    }

    // Method to calculate net salary based on basic salary
    public static double calculateNetSalary(double
basicSalary) {
        double netSalary = 0;

        if (basicSalary < 3000) {
            netSalary = basicSalary + (basicSalary * 0.02);
        } else if (basicSalary >= 3000 && basicSalary < 5000)
{
```

```
            netSalary = basicSalary + (basicSalary * 0.05);
        } else if (basicSalary >= 5000 && basicSalary < 10000)
{
            netSalary = basicSalary + (basicSalary * 0.10);
        } else {
            netSalary = basicSalary + (basicSalary * 0.20);
        }

        return netSalary;
    }
}
```

17. PRAC NO.17

```
import java.util.Scanner;

public class DiscountCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        // Input credit card type

        System.out.print("Enter credit card type
(VISA/XYZ/ABC): ");

        String creditCardType =
scanner.nextLine().toUpperCase();


        // Input shopping amount

        System.out.print("Enter shopping amount: ");

        double shoppingAmount = scanner.nextDouble();


        // Calculate discount based on credit card type and
shopping amount

        double discount = calculateDiscount(creditCardType,
shoppingAmount);


        // Calculate net price

        double netPrice = shoppingAmount -
(shoppingAmount * discount / 100);


        // Display net price
        System.out.println("Net Price (Rs.): " + netPrice);

        scanner.close();
    }

    // Method to calculate discount based on credit card type and shopping amount
    public static double calculateDiscount(String creditCardType, double shoppingAmount) {
        double discount = 0;

        switch (creditCardType) {
            case "VISA":
                discount = shoppingAmount < 5000 ? 10 : 20;
                break;
            case "XYZ":
                discount = shoppingAmount < 10000 ? 15 : 25;
                break;
            case "ABC":
                discount = shoppingAmount < 8000 ? 12 : 15;
                break;
            default:
                System.out.println("Invalid credit card type!");
        }

        return discount;
    }
}
```

18. PRAC NO.27

```
class Employee {
    // Data members
    int empNo;
    String empName;
    String job;
```

```
        // Method to display employee details
        void showInfo() {
                System.out.println("Employee Number: " + empNo);
                System.out.println("Employee Name: " +
empName);
                System.out.println("Job: " + job);
        }
}

class Salary extends Employee {
        // Additional variables
        float basic;
        double newSal;

        // Method to calculate new salary
        void calculate(double perc) {
                newSal = basic + (basic * perc / 100);
        }
```

```
    // Method to display basic salary
    void dispData() {
        System.out.println("Basic Salary: " + basic);
    }
}
```

31. Write a Java program to create a class student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age is not between 15 and 21 then generate a user defined exception “Age not within the range”

```
class AgeNotWithinRangeException extends Exception {
	public AgeNotWithinRangeException(String
message) {
		super(message);
	}
}

class Student {
	private int rollNo;
	private String name;
	private int age;
	private String course;

	public Student(int rollNo, String name, int age,
String course) throws AgeNotWithinRangeException {
        if (age < 15 || age > 21) {
            throw new AgeNotWithinRangeException("Age not within the range (15-21)");
        }
        this.rollNo = rollNo;
        this.name = name;
        this.age = age;
        this.course = course;
    }

    public void display() {
        System.out.println("Roll No: " + rollNo);
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
        System.out.println("Course: " + course);
    }
}


public class Main {
    public static void main(String[] args) {
        try {
            Student student1 = new Student(1,
"John", 18, "Computer Science");
            student1.display();


            // This will throw
AgeNotWithinRangeException
            Student student2 = new Student(2,
"Alice", 22, "Electrical Engineering");
            student2.display();
        } catch (AgeNotWithinRangeException e) {
            System.out.println("Exception: " +
e.getMessage());
```

```
            }
        }
}
```

32. Write a Java AWT program to sort the array in ascending order

```
import java.awt.*;
import java.awt.event.*;
import java.util.Arrays;

public class SortArray extends Frame implements
ActionListener {
        TextArea unsortedArrayArea, sortedArrayArea;
        Button sortButton;
        int[] array = {5, 3, 8, 1, 4};
```

```
    SortArray() {
        setLayout(new FlowLayout());

        unsortedArrayArea = new TextArea("Unsorted Array:\n" + Arrays.toString(array), 5, 40);
        add(unsortedArrayArea);

        sortButton = new Button("Sort Array");
        add(sortButton);
        sortButton.addActionListener(this);

        sortedArrayArea = new TextArea("Sorted Array:\n", 5, 40);
        add(sortedArrayArea);

        setSize(300, 300);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e) {

        Arrays.sort(array);

        sortedArrayArea.setText("Sorted Array:\n" +
Arrays.toString(array));

    }


    public static void main(String[] args) {

        new SortArray();

    }

}
```

33. Write a Java code to design the following GU

Error occurring

```
import java.applet.Applet;
import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;


public class SimpleCalculatorApplet extends Applet {

        TextField firstNumberField;

        TextField secondNumberField;

        TextField resultField;

        Choice operationChoice;


        public void init() {

                Label firstNumberLabel = new Label("Enter 1st no.");

                Label secondNumberLabel = new Label("Enter 2nd no.");

                Label resultLabel = new Label("Result");


                firstNumberField = new TextField();

                secondNumberField = new TextField();

                resultField = new TextField("0");
                resultField.setEditable(false);

                operationChoice = new Choice();
                operationChoice.add("+");
                operationChoice.add("-");
                operationChoice.add("*");
                operationChoice.add("/");

                add(firstNumberLabel);
                add(firstNumberField);
                add(secondNumberLabel);
                add(secondNumberField);
                add(resultLabel);
                add(resultField);
                add(operationChoice);

                operationChoice.addActionListener(e ->
actionPerformed(e));
        }
        public void actionPerformed(ActionEvent e) {
                double result = 0.0;
                double firstNumber =
Double.parseDouble(firstNumberField.getText());
                double secondNumber =
Double.parseDouble(secondNumberField.getText());

                String operation =
operationChoice.getSelectedItem();
                switch (operation) {
                    case "+":
                        result = firstNumber + secondNumber;
                        break;
                    case "-":
                        result = firstNumber - secondNumber;
                        break;
                    case "*":
                        result = firstNumber * secondNumber;
                        break;
                    case "/":
                        if (secondNumber != 0) {
                            result = firstNumber / secondNumber;
                        } else {
                            resultField.setText("Error: Division by
zero");
                            return;
                        }
                        break;
                }
                resultField.setText(String.valueOf(result));
        }
}
```

34. Write a Java program to read the student data from user and store it in the file.

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class StudentDataToFile {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter student name: ");
        String name = scanner.nextLine();

        System.out.print("Enter student ID: ");
        int id = scanner.nextInt();
        System.out.print("Enter student age: ");
        int age = scanner.nextInt();

        // Create a string with the student data
        String studentData = "Name: " + name + ", ID: " + id + ", Age: " + age;

        // Write the student data to a file
        try (BufferedWriter writer = new BufferedWriter(new FileWriter("student_data.txt"))) {
            writer.write(studentData);
            System.out.println("Student data has been written to student_data.txt");
        } catch (IOException e) {
            System.err.println("Error writing to file: " + e.getMessage());
        }
```

```
    }
}
```