College of Arts, Science & Commerce (Autonomous)

SIES

RISE WITH EDUCATION

NAAC REACCREDITED - 'A' GRADE

S.I.E.S College of Arts, Science and Commerce
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that Mr. /~~Miss~~. **Namal Lingraj**

Roll No. **SCS2223054** has successfully completed the necessary course of experiments in the subject of **Core Java** during the academic year **2022 – 2023** complying with the requirements of **University of Mumbai**, for the course of **S.Y. BSc. Computer Science [Semester-3]**

Prof. In-Charge
Prof. Shivani Deopa
(Core Java)

Examination Date:
Examiner's Signature & Date:

Head of the Department
Prof. Manoj Singh

College Seal
And
Date

CORE JAVA JOURNAL

PRACTICALS INDEX

NO	AIM	SIGN
1	Accept integer values for a, b, and c which are coefficients of quadratic equation. Find the solution of quadratic equation.	
2	Accept two n * m matrices. Write a Java program to find addition of these matrices.	
3	Accept n strings. Sort names in ascending order.	
4	Create a package: Animals. In package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package Animals.	
5	Demonstrate Java inheritance using extends keyword.	
6	Demonstrate method overloading and method overriding in Java.	
7	Demonstrate creating your own exception in Java.	
8	Using various AWT components design Java application to accept a student's resume. (Design form)	
9	Write a Java List example and demonstrate methods of Java List interface.	
10	Design simple calculator GUI application using AWT components.	

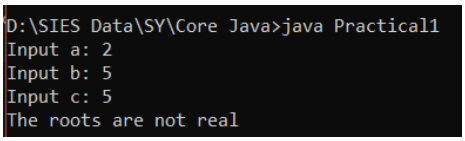
Practical 1:

Code:

```
import java.util.Scanner;
public class Practical1
{
  public static void main(String[] Strings)
{
Scanner input=new Scanner(System.in);
System.out.print("Input a: ");
double a=input.nextDouble();
System.out.print("Input b: ");
double b=input.nextDouble();
System.out.print("Input c: ");
double c=input.nextDouble();

double result=b*b-4.0*a*c;

if(result>0.0)
{
double r1=(-b+Math.pow(result,0.5))/(2.0*a);
double r2=(-b-Math.pow(result,0.5))/(2.0*a);
System.out.print("The first root is" +r1+" and the second root is " +r2);
}
else if (result==0.0)
{
double r3=-b/(2.0*a);
System.out.print("The root is" +r3);
```

```
}
else
{
System.out.println("The roots are not real");
}
}
}
```

Output:



Practical 2

Code:

```
class Matrixadd
{
public static void main(String args[])
{
int m,n,i,j,k=2;
m=Integer.parseInt(args[0]);
n=Integer.parseInt(args[1]);
int a[][]=new int[m][n];
int b[][]=new int[m][n];
int c[][]=new int[m][n];
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
a[i][j]=Integer.parseInt(args[k]);
k++;
}}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
b[i][j]=Integer.parseInt(args[k]);
k++;
}}
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
```

```
{
c[i][j]=a[i][j]+b[i][j];
}}
System.out.println("First matrix");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
System.out.print(a[i][j]+" ");
}
System.out.println();
}
System.out.println("Second Matrix");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
System.out.print(b[i][j]+" ");
}
System.out.println();
}
System.out.println("Result MAtrix AFter Addition");
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
System.out.print(c[i][j]+ " ");
}
System.out.println();
}
}
}
```

Output:

```
D:\SIES Data\SY\Core Java>java Matrixadd 1 2 3 4 5 6 7 8 9
First matrix
3 4
Second Matrix
5 6
Result MAtrix AFter Addition
8 10
```

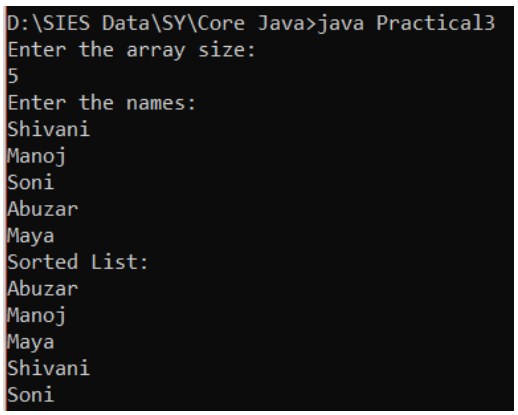
Practical 3

Code:

```
import java.io.*;
import java.util.Scanner;
class Practical3{
   public static void main(String[]  args)
{
  Scanner sc=new Scanner(System.in);
System.out.println("Enter the array size:");
int n=sc.nextInt();
String names[]=new String[n];

System.out.println("Enter the names:");
for(int i=0;i<n;i++)
{
       names[i]=sc.next();
}
String temp;
for (int i=0;i<n;i++){
    for(int j=i+1;j<n;j++){

     if (names[i].compareTo(names[j])>0)
     {
      temp=names[i];
      names[i]=names[j];
      names[j]=temp;
         }
      }
}

System.out.println("Sorted List:");
```

```
for(int i=0;i<n;i++){
System.out.println(names[i]);
}
}}
```

Output:



Practical 4

Code:

```
package Animal;
public interface Behaviour
{
public void eat();
public void type();
}
import Animal.*;
public class Dolphin implements Behaviour
{
public void eat()
{
 System.out.println("Dolphin eats small fishes and weeds");
}
public void type()
{
 System.out.println("Dolphin are omnivorous");
}
public static void main(String args[])
{
 Dolphin e=new Dolphin();
 e.eat();
 e.type();
}
}
```

Output:

```
D:\SIES Data\SY\Core Java>java Dolphin
Dolphin eats small fishes and weeds
Dolphin are omnivorous
```

Practical 5

Code:

```
import java.util.*;
class AddSub
{
public void Add(int a,int b)
{
int c = a+b;
System.out.println("The Addition is : "+c);
}
public void Sub(int a,int b)
{
int c = a-b;
System.out.println("The Subtraction is : "+c);
}
}
class MulDiv extends AddSub
{
public void Mul(int a,int b)
{
int c = a*b;
System.out.println("The Multiplication is : "+c);
}
public void Div(int a,int b)
{
int c = a/b;
System.out.println("The Division is : "+c);
}
}
```

```
class Inheritance
{
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter number1: ");
int a = sc.nextInt();
System.out.println("Enter number2: ");
int b = sc.nextInt();
MulDiv md=new MulDiv();
md.Add(a,b);
md.Sub(a,b);
md.Mul(a,b);
md.Div(a,b);
}
}
```

Output:

```
D:\SIES Data\SY\Core Java>java Inheritance
Enter number1:
12
Enter number2:
10
The Addition is : 22
The Subtraction is : 2
The Multiplication is : 120
The Division is : 1
```

Practical 6

Overloading Code:

```
import java.io.*;
  class Overloading{
      public static void main(String[] args) {


          Geometry obj = new Geometry();
          obj.Area(2.5);
          obj.Area(2);
          obj.Area(2,4);
    }
  }
 class Geometry {
     double PI = 3.14;
     void Area(double r){
         double A = PI * r * r;
         System.out.println("Area of the circle is : " + A);
     }
     void Area(int s){
         double A = s* s;
         System.out.println("Area of the square  is : " + A);
     }void Area(double l, double b){
         double A = l * b;
         System.out.println("Area of the rectangle is : " + A);
     }
}
```

Overriding Code:

```
import java.util.*;

class Vehicle
{
 void VehicleName()
 {
  System.out.println("Aeroplane");
 }
 void VehicleModel()
 {
  System.out.println("A504");
 }
}

class Car extends Vehicle
{
 void VehicleName()
 {
  System.out.println("Audi");
 }
 void VehicleModel()
 {
  System.out.println("X500");
 }
}

class Bike extends Vehicle
{
 void VehicleName()
 {
```

```
  System.out.println("Passion");
 }
 void VehicleModel()
 {
  System.out.println("RS200");
 }
}


class Overriding
{
 public static void main(String[] args)
 {
  Vehicle v = new Vehicle();
  Car c = new Car();
  Bike b = new Bike();

  v.VehicleName();
  v.VehicleModel();
  c.VehicleName();
  c.VehicleModel();
  b.VehicleName();
  b.VehicleModel();
 }
}
```

Output:

```
D:\SIES Data\SY\Core Java>java Overloading
Area of the circle is : 19.625
Area of the square  is : 4.0
Area of the rectangle is : 8.0
```

```
D:\SIES Data\SY\Core Java>java Overriding
Aeroplane
Boeing 777
Ferrari
458 Spider
Ninja
H2R
```

Practical 7

Code:

```
import java.util.Scanner;



class NotProperNameException extends RuntimeException
    {
        NotProperNameException(String msg){
            super(msg);
        }
    }
public class CustomCheckedException{
    String name;
    int age;
    public static boolean containAlphabet(String name){
        for (int i = 0; i < name.length();i++){
            char ch = name.charAt(i);
            if (!(ch >= 'a' && ch <= 'z')){
                return false;
            }
        }
        return true;
    }
    public CustomCheckedException(String name, int age){
        if(!containAlphabet(name) && name!=null){
            String msg = "Improper name (Should contain only characters between a to z (all small))";
            NotProperNameException exName = new NotProperNameException(msg);
            throw exName;
        }
        this.name = name;
        this.age = age;
```

```
    }
    public void display(){
        System.out.println("Name of the Student: "+this.name);
        System.out.println("Age of the Student: "+this.age);
    }
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the name of the person: ");
        String name = sc.next();
        System.out.println("Enter the age of the person: ");
        int age = sc.nextInt();
        CustomCheckedException obj = new CustomCheckedException(name,age);
        obj.display();
    }
}
```

Output:

```
D:\SIES Data\SY\Core Java>java CustomCheckedException
Enter the name of the person:
lingraj
Enter the age of the person:
19
Name of the Student: lingraj
Age of the Student: 19
```

```
D:\SIES Data\SY\Core Java>java CustomCheckedException
Enter the name of the person:
Lingraj
Enter the age of the person:
19
Exception in thread "main" NotProperNameException: Improper name (Should contain only characters between a to z (all small)
)
        at CustomCheckedException.<init>(CustomCheckedException.java:25)
        at CustomCheckedException.main(CustomCheckedException.java:41)
```

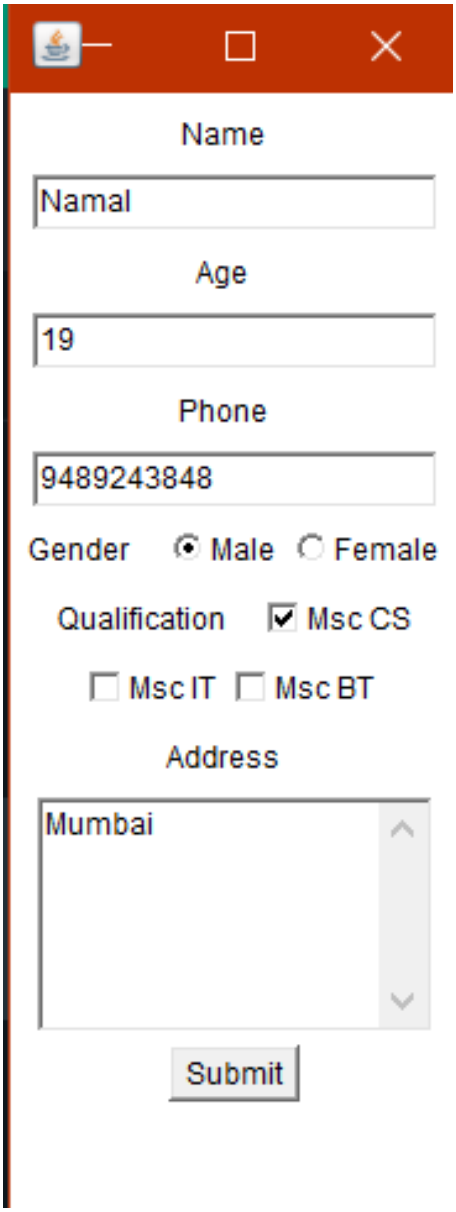
Practical 8

Code:

```
import java.awt.*;
public class Practical8 extends Frame{
    TextField txtName, txtAge, txtPhone;
    Checkbox chkMale, chkFemale;
    Checkbox chkQ1, chkQ2, chkQ3;
    CheckboxGroup cbgGender;
    Button btnSubmit;
    TextArea txaAddress;
    public Practical8(){

        txtName = new TextField(20);
        txtAge = new TextField(20);
        txtPhone = new TextField(20);

        cbgGender = new CheckboxGroup();
        chkMale = new Checkbox("Male", false, cbgGender);
        chkFemale = new Checkbox("Female", false, cbgGender);
        chkQ1 = new Checkbox("Msc CS");
        chkQ2 = new Checkbox("Msc IT");
        chkQ3 = new Checkbox("Msc BT");

        txaAddress = new TextArea(5, 20);

        btnSubmit = new Button("Submit");

        add(new Label("Name"));
        add(txtName);
        add(new Label("Age"));
```

```
        add(txtAge);
        add(new Label("Phone"));
        add(txtPhone);
        add(new Label("Gender"));
        add(chkMale);
        add(chkFemale);
        add(new Label("Qualification"));
        add(chkQ1);
        add(chkQ2);
        add(chkQ3);
        add(new Label("Address"));
        add(txaAddress);
        add(btnSubmit);
    }

    public static void main(String[] args) {
        Practical8 practical8 = new Practical8();
        practical8.setLayout(new FlowLayout());
        practical8.setSize(200,500);
        practical8.setVisible(true);
    }

}
```

Output:


Name
Namal
Age
19
Phone
9489243848
Gender
Male
Female
Qualification
Msc CS
Msc IT
Msc BT
Address
Mumbai
Submit

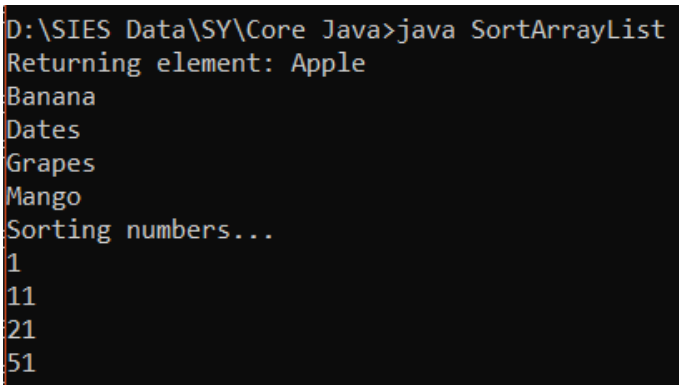
Practical 9

Code:

```
import java.util.*;
class SortArrayList{
public static void main(String args[]){
List <String> list1=new ArrayList<String>();
list1.add("Mango");
list1.add("Apple");
list1.add("Banana");
list1.add("Grapes");
System.out.println("Returning element: "+list1.get(1));
list1.set(1,"Dates");
Collections.sort(list1);
for(String fruit:list1)
System.out.println(fruit);
System.out.println("Sorting numbers...");

List<Integer> list2=new ArrayList<Integer>();
list2.add(21);
list2.add(11);
list2.add(51);
list2.add(1);
Collections.sort(list2);
for(Integer number:list2)
System.out.println(number);
```

```
}
}
```

Output:


D:\SIES Data\SY\Core Java>java SortArrayList
Returning element: Apple
Banana
Dates
Grapes
Mango
Sorting numbers...
1
11
21
51

Practical 10

Code:

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

class Calculator  implements ActionListener
{
        Frame f = new Frame();
        Label l1 = new  Label("Enter first Number");
        Label l2 = new Label("Enter second Number");
        Label l3 =new  Label("Result");
        TextField t1 = new TextField();
        TextField t2 = new TextField();
        TextField t3 = new TextField();
        Button b1 = new Button("Add");
        Button b2 = new Button("Sub");
        Button b3 = new Button("Mul");
        Button b4 = new Button("Div");
Calculator(){
                l1.setBounds(50,100,150,20);
                l2.setBounds(50,150,150,20);
                l3.setBounds(50,200,150,20);
                t1.setBounds(250,100,100,20);
                t2.setBounds(250,150,100,20);
                t3.setBounds(250,200,100,20);
                b1.setBounds(50,250,50,20);
```

```
        b2.setBounds(110,250,50,20);
        b3.setBounds(170,250,50,20);
        b4.setBounds(230,250,50,20);
        f.add(l1);
        f.add(l2);
        f.add(l3);
        f.add(t1);
        f.add(t2);
        f.add(t3);
        f.add(b1);
        f.add(b2);
        f.add(b3);
   f.add(b4);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        f.setLayout(null);
        f.setVisible(true);
        f.setSize(500,500);

}

public void actionPerformed(ActionEvent e){
        int i = Integer.parseInt(t1.getText());
        int j = Integer.parseInt(t2.getText());

        if(e.getSource()==b1){
                t3.setText(String.valueOf(i+j));
        }
        if(e.getSource()==b2){
                t3.setText(String.valueOf(i-j));
```

```
                }
                if(e.getSource()==b3){
                        t3.setText(String.valueOf(i*j));
                }
                if(e.getSource()==b4){
                        t3.setText(String.valueOf(i/j));
                }
        }

        public static void main(String args[]){
                new Calculator();
        }
}
```

Output: