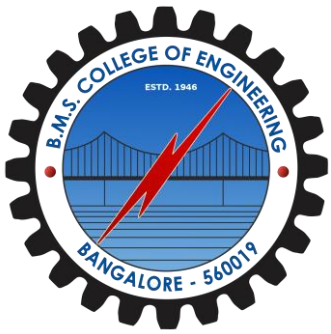
BMS COLLEGE OF ENGINEERING

(Autonomous College under VTU)
Bull Temple Road, Basavanagudi, Bangalore-560019
2022-23



LAB REPORT

Of

"All Programs Executed in Java"

In

Object Oriented Programming using JAVA (OOJ)

By

RISHABH KUMAR

(1BM22CS221)

CSE - 3D

Submitted To

Dr. SEEMA PATIL
Assistant Professor
Department of Computer Science and Engineering

OOS Lab - 1

12/12/23

(Q) A java program to print all the real solutions.

```
import java.util.Scanner;
class Quadratic
{
  int a,b,c;
  double r1,r2,d;
  void getd()
    {
      Scanner s = new Scanner (System.in);
      System.out.println (" Enter the coefficient of a,b,c");
      a = s.nextInt();
      b = s.nextInt();
      c = s.nextInt();
    }
  void compute()
    {
      while (a==0)
        {
          System.out.println ("Not a quadratic equation");
          System.out.println ("Enter a non zero value of a:");
          Scanner s = new Scanner (System.in);
          a = s.nextInt();
        }
      d = b*b - 4*a*c;
      if (d==0)
        {
          r1 = (-b)/(2*a)
```

```
System.out.println("Roots are real and equal");
System.out.println("Root1 = Root2 =" +r1);
}
else if (d>0)
{
r1 = ((-b) + (Math.sqrt(d))) / (double) (2*a);
r2 = ((-b) - (Math.sqrt(d))) / (double (2*a);
System.out.println("Roots are real and
distinct");
System.out.println("Root1 =" +r1+"
Root2 =" +r2);
}
else if (d<0)
{
System.out.println("Roots are imaginary");
r1 = (-b) / (2*a);
r2 = Math.sqrt(-d) / (2*a);
System.out.println("Root1 =" +r1+
" + i" +r2);
System.out.println("Root1 =" +r1+"-i"
+r2);
}
}
}
```

```
class Quadratic Main
{
    public static void main (String args[])
    {
        System.out.print ("Rishabh, 1BM22CS221");
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

(Q) A java program to find the area of the rectangle.

```
class RectangleArea {
    public static void main (String args[]){
        int length, breadth;
        length = Integer.parseInt (args[0]);
        breadth = Integer.parseInt (args[1]);
        int area = length * breadth;
        System.out.println ("Rishabh, 1BM22CS221");
        System.out.println ("area =" + area);
    }
}
```

(Q) A java program to find the factorial of a number

```
class factorial {
    public static void main (String args[])
    {
      int fac = 1;
      System.out.println("Enter a number: ");
      Scanner sc = new Scanner (system.in);
      int n = sc.nextInt();
      for (int i = 1 ; i<=n ; i++){
            fac = fac*i ;
      }
      System.out.println("Rishabh , 1BM22CS221");
      System.out.println ("Factorial:" + fac);
    }
}
```

(Q) To find the given 5 digit int is palindrome or not.

```
class palindrome {
    public static void main (String args[])
    {
     int n, t, rem, rev = 0;
     Scanner sc = new Scanner (System.in);
     System.out.println ("Enter a 5 digit numbers");
     n = sc.nextInt();
```

```
t=n;
while (t>0){
    rem = t%10;
    rev = rev*10 + rem;
    t = t/10;
    }
    if (rev == n){
      System.out.println("Palindrome");
      }
    else {
      System.out.println("not Palindrome");
}
}
}
```

(Q) To check whether the number is prime or not.

```
import java.util.*;
class isprime{
static void isprime(int n){
    int i, m=0, flag=0;
    m = n/2
    if (n==0||n==1){
      System.out.println(n + " is not a prime number");
}
}
```

```
else {
    for (i=2; i<m; i++) {
        if (n%i==0) {
            System.out.println(n+" is not a prime no.");
            break;
        }
    }
    if (flag==0) {
        System.out.println(n+" is a prime");
    }
}
}

public static void main (String args[]) {
    int i;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter i:");
    i = sc.nextInt();
    isprime(i);
}
}
```

(Q) Find sum of digits in a number.

```
class Sum of digits {
    public static void main (String args[]) {
        long number, sum;
        Scanner sc = new Scanner (System.in);
        System.out.println ("Enter no.:");
        number = sc.nextLong();

    for (sum = 0; number != 0; number /= 10) {
        sum = sum + number % 10
    }
    System.out.println ("Sum of digits" + sum);
}
}
```

(Q) Find the greatest number.

```
class largestonemethod {
    static void largest (int i, int j, int k) {
        if (i >= j && i >= k) {
            System.out.println (i + " is largest");
        }
        if (j >= i && j >= k)
            System.out.println (j + " is the largest");
    }
```

02/01/24

OOJ Lab - 2

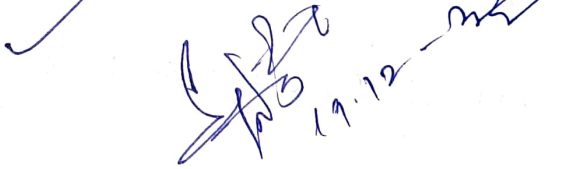
19th Dec 2023

(Q) To Calculate the SGPA

```
import java.util.Scanner;

class Student {
    private String usn;
    private String name;
    private int[] marks;
    private int[] credits;

    public Student() {
        usn = "";
        name = "";
    }

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

    System.out.print("Dishath Kumar, 1BM22CS221");
    System.out.print("Enter USN: ");
    usn = scanner.nextLine();

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

System.out.print("Enter number of subjects");
int numSubjects = scanner.nextInt();

credits = new int[numSubjects];
marks = new int[numSubjects];
```

```
for (int i=0 ; i < credits.length ; i++) {
    System.out.println ("Subject" + (i+1) + "- credits"
        + credits[i] + "Marks:" + marks[i]);
}
}
public double calculateSGPA () {
    double totalCredits = 0;
    double totalGradePoints = 0;
for (int i=0 ; i < credits.length ; i++) {
    totalCredits += credits[i];
    totalGradePoints += calculateGradePoints (marks[i])
                        * credits[i];
    return totalGradePoints / totalCredits;
}
private double calculateGradePoints (int marks) {
    if (marks >= 90) {
        return 10.0;
    }
    else if (marks >= 80) {
        return 9.0;
    }
    else if (marks >= 70) {
        return 8.0;
    }
    else if (marks >= 60) {
        return 7.0;
    }
}
```

```
        else if (marks >= 60){
            return 5.0;
        }
        else{
            return 0.0;
        }
    }
}

public class Main {
    public void main(String [] args ){
        Student student = new Student();
        Student.acceptDetails();
        Student.displayDetails();
        double sgpa = Student.calculateSGPA();
        System.out.println("SGPA: " + sgpa);
    }
}
```



Sample Output:

Rishabh, 1BM22CS224

Enter USN: ABC123

Enter Name: John Doe

Enter no of subjects: 3

Enter credits for subject 1: 4

Enter marks for subject 1: 85

Enter credits for subject 2 : 3

Enter marks for subject 2 : 25

Enter credits for subject 3 : 5

Enter mark for subject 3 : 90

SGPA : 8.433333334

Rishabh Kumar

1BM22CS221

OOJ Lab-3

26/12/23

(1) To store the details of books.

```
class Book {
    private String name;
    private String author;
    private double price;
    private int numPages;

public Book (String name, String author, double price,
                int numPages) {
    this.name = name;
    this.author = author;
    this.price = price;
    this.numPages = numPages;
}

public String getName() {
    return name;
}

public void setName (String name) {
    this.name = name;
}

public String getAuthor () {
    return author;
}

public void setAuthor (String author) {
    this.author = author;
}
```

```
System.out.println ("Rishabh, 1BM22CS221");

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

Output :-

```
Rishabh, 1BM22CS221

Book Details:
Name: Catcher
Author: J.D.
Price: $15.99
Number of Pages: 224

Book Details:
Name: Kill
Author: Lee
Price: $12.5
Number of Pages: 536
```

26.12.23

OOS - Lab 4

02/01/24

(Q) To make a class Shape and execute the given instructions.

```
abstract class Shape {
    protected int dim1;
    protected int dim2;

    public Shape (int dim1, int dim2) {
        this.dim1 = dim1;
        this.dim2 = dim2;
    }

    public abstract void printArea();
}

class Rectangle extends Shape {
    public Rectangle (int len, int wid) {
        super (len, wid);
    }

    public void printArea () {
        int area = dim1 * dim2;
        System.out.println ("Area of Rectangle:" + area);
    }
}

class Triangle extends Shape {
    public Triangle (int base, int height) {
        super (base, triangle);
    }
```

```
public void printArea() {
double area = 0.5 * dim1 * dim2;
System.out.println("Area of Triangle: " + area);
}
}

class Circle extends Shape {
    public Circle(int rad) {
        super(rad, 0);
    }

    public void printArea() {
double area = Math.PI * dim1 * dim2;
System.out.println("Area of Circle: " + area);
}
}

public class Main {
    public static void main(String args[]) {
Rectangle rectangle = new Rectangle(5, 10);
Triangle triangle = new Triangle(4, 6);
Circle circle = new Circle(7);

rectangle.printArea();
triangle.printArea();
circle.printArea();
}
}
```

Sample Output

Area of Rectangle: 50

Area of triangle : 12.0

Area of circle : 153.93804002589985

Rishabh Kumar
1BM22CS224

02/01/24

OOJ Lab-5 9/01/24

```
import java.util.Scanner;

class Account {

    String customerName;
    int accountNumber;
    String accountType;
    double Balance;

    public Account(String customerName, int accountNumber,
                   String accountType, double balance){

        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

public void displayBalance(){

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

class SavingsAccount extends Account{

    double interestRate;

public SavingsAccount(String customerName, int
    accountNumber, String AccountType, double Balance, double
    interestRate){

        super(customerName, accountNumber, accountType,
                balance);
    this.interestRate = interestRate;
}
```

```
public void computeInterest () {
    balance += (balance * interestRate) / 100;
    System.out.println ("Interest Rate added");
}

public void withdraw (double amount) {
    if (amount <= balance) {
        balance -= amount;
        System.out.println ("withdraw successful");
    }
    else {
        System.out.println ("Insufficient Fund");
    }
}
}

class CurrentAccount extends Account {
    double minBalance;
    double serviceCharge;

public CurrentAccount (String customerName, int
accountNumber, String accountType, double balance,
double minBalance, double serviceCharge) {

    super (customerName, accountNumber, accountType,
    balance);

    this.minBalance = minBalance;
    this.serviceCharge = serviceCharge;
}
```

```
public void withdraw(double amount){
 if (balance - amount >= minBalance){
     balance -= amount;
     System.out.println("Withdraw Successful");
 }
 else{
 System.out.println("Insufficient Fund");
 balance -= serviceCharge;
}
}
}

public class Bank{

    public static void main(String args[]){
    SavingsAccount savings = new SavingAccount
      ("John", 001, "Savings", 5000, 5);
    CurrentAccount current = new CurrentAccount
      ("Alice", 2001, "Current", 8000, 100, 10);

 savings.computeInterest();
 Savings.displayBalance();
 Savings.withdraw(2000);
 Savings.displayBalance();

 current.displayBalance();
 Current.withdraw(5000);
 current.displayBalance();

}

}
```

Sample Output :-

Interest rate added
Account Balance : 5250.0
Withdrawal Successful
Account Balance : 3250.0
Account Balance : 3000.0
Withdrawal Successful
Account Balance : 3000.0

Rishabh Kumar
1BM22CS224

09.01.24

Rishabh Kumar
1BM22CS224

OOJ Week ~~Lab~~ – 6

16/01/24

(String Additional Programs)

① • Created a String using a literal.
• Created a String using a new keyword.
• Created a String using StringBuffer.

② String literal : Hello
String length : 7
String concatenation: Hello, World!

③ Original Number: 42
toString of number : 42
Original float : 3.14
toString of float: 3.14
Original Boolean : true
toString of Boolean : true

④ Extracted SubString : BMSCE

⑤ Byte Array: 74 101 108 111 42 32 111
char Array: J a v a i$ f o n !

⑥ Bmsce equals Bmsce → true
Bmsce equals College → false
Bmsce equals BMSCE → false
Bmsce equalIgnoreCase BMSCE → true

(7) Substring is matched

(8) Starts with "Hello": true
Starts with "World": false

(9) Ends with "World": true
Ends with "Hello": false

(10) Using equals(): true
Using ==: false

(11) Sorted words: apple ball cat dog ant free
gun hen ice jug kite tgt
mon net orange parrot queen
sing star tree umbrella van
watch xmas yatch zee

(12) Sorted Numbers: 1 2 3 4 5 6 7
8 9 10

(13) Modified String: "This is a test, This is, too."

(14) Concatenated String: helloworld

(15) Modified String: Welcome to Bmsce commege
of Enggineering

(16) Original String: " Hello Friends "
Trimmed String: 'Hello Friends'

(17) Original Student Records:

Registration Number : 101

Full Name : John Doe

Semester : 3

CGPA : 8.3

(18) setLength() : Hello

charAt(1) : e

setCharAt(1, 'x') : HxllO

getChars() : HxllO

append : HxllO Java

insert(6, 'Awesome') : HxllO Awesome Java

(19) Eagle is flying high in the sky.

Eagle makes a screeching sound.

Hawk is gliding gracefully.

Hawk makes a sharp cry.

(20) Circle Area : 78.53981633974483

Circle Perimeter : 31.415926535897933

Triangle Area : 6.0

Triangle Parameter : 12.0

18.01-m

OOJ Week-7 (Lab-6)

23/01/24

~~Internals.java~~ Main.java

```
import CIE.Internals;
import SEE.External;

public class CalculateFinalMarks {

    Internal student1 = Internals("123", "John Doe", 5,
                        new int[]{85, 90, 78, 92, 88});

    External student1 = External("189", "Rock", 5,
                        new int[]{95, 60, 98, 92, 68});

    System.out.println("Student 1 - Final Marks:"
                        + finalMarks1);
    System.out.println("Student 2 - Final Marks:"
                        + finalMarks2);

}

public static int calculateFinalMarks(int[] marks){
    int totalMarks = 0;
    for(int mark: marks){
        totalMarks += mark;
    }

    return totalMarks;

}

}
```

Student.java

```
package CIE;

public class Student {

    String usn, name;
    int sem;

    public Student (String usn, String name, int sem){
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

Internals.java

```
package CIE;

public class Internals extends Student {

    int [] internalMarks;

    public Internals (String usn, String name,
        int sem, int [] internalMarks ) {
        super (usn, name, sem);
        this.internalMarks = internalMarks;
    }

}
```

External.java

```
package SEE;

import CIE.Student;

public class External extend Student {
        int[] seeMarks;

    public External(String usn, String name, int sem,
    int seeMarks[]) {

        super(usn, name, sem);
        this.seeMarks = seeMarks;
    }
}
```

Output :-

```
Rishabh Kumar
1BM22CS224
Student 1 - Final Marks : 433
Student 2 - Final Marks : 427
```

23.01.24

OOJ Week-8 (Lab-7)

30/01/24

(8) Create a base class father and a derived class son and execute according the given conditions.

```
import java.util.Scanner;
class WrongAge extends Exception {
    WrongAge(String message) {
        super(message);
    }
}
class InputScanner {
    static Scanner scanner = new Scanner(System.in);
}
class Father extends InputScanner {
    int fatherAge;
    Father() throws WrongAge {
        System.out.println("Enter Father age : ");
        fatherAge = scanner.nextInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}
```

```
void display() {
    System.out.println("Father's Age: " + fatherAge);
  }
}

class Son extends Father {
    int sonAge;

  Son() throws WrongAge {
      System.out.println("Enter Son's Age: ");
      sonAge = scanner.nextInt();
    if (sonAge > fatherAge) {
    throw new WrongAge("Son's Age cannot be
    greater than father's Age");
    }
    else if (sonAge < 0) {
  throw new WrongAge("Age cannot be negative");
    }
  }

void display() {
    super.display();
    System.out.println("Son's Age: " + sonAge);
  }
}

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

```
Son son = new Son();
son.display();
}
catch (WrongAge e){
    System.out.println("Exception: " + e.getMessage());
}
}
}
```

Output :-

Enter Father's Age : 40

Enter Son's Age : 45

Exception : Son's age cannot be greater the father's Age

Rishabh Kumar

1BM22CS224

30.01.24

(Lab-8) 06-02-24

(Q) Executing Threading with an Example.

```
class DisplayThread extends Thread {
    private String message;
    private int sleepTime;

    public DisplayThread (String message, int sleepTime) {
        this.message = message;
        this.sleepTime = sleepTime;
    }

    public void run () {
        while (true) {
            System.out.println (message);
            try {
                Thread.sleep (sleepTime * 1000);
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class Main {
    public static void main (String [] args) {
        var thread1 = new DisplayThread
            ("BMSCE", 10);
```

```
var thread2 = new DisplayThread("CSE", 2);
thread1.start();
thread2.start();
}
}
```

Output:-

BMSCE
CSE
CSE
CSE
Rishabh Kumar
1BM22CS224

OOJ Week-10

(lab-10)

```
public class DeadlockExample {
    public static void main (String [] args) {
        final SR sr = new SR ();
        Thread one = new Thread (() -> {
            try {
                sr.method1();
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
        Thread two = new Thread (() -> {
            try {
                sr.method2();
            }
            catch (InterruptedException e) {
                e.printStackTrace();
            }
        });
        one.start();
        two.start();
    }
}
```

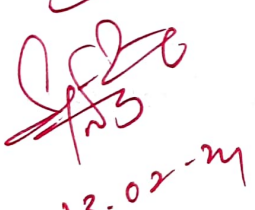
```
class SR {
    private final Object lock1 = new Object();
    private final Object lock2 = new Object();

public void method1() throws InterruptedException {
    Synchronized(lock1) {
        System.Out.println("method 1 acquired
                            lock1");
        Thread.sleep(1000);
    Synchronized (lock2) {
        Systemout.println("Method 1 acquired
                           lock2");
    }
}

public void method2() throws Interrupted {
    Synchnored (lock2) {
        System.out.println("Method 2");
        Thred.sleep(1000);

    Synchronized(lock1) {
        System.out.println("method 2");
    }
}
}
```

Output :-

Method 1 acquired lock 1
Method 2 acquired lock 2

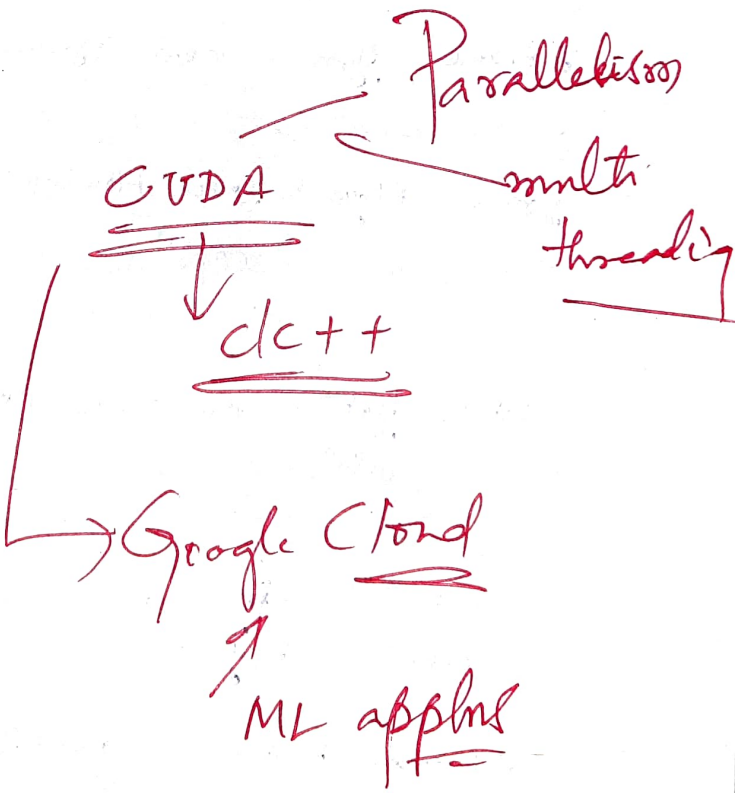
06.02.24

```
public class DeadlockExample {
    public static void main (String[] args) {
        final SR sr = new SR();
        Thread one = new Thread();
        try{
            sr.method1();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```



13.02.24

Rishabh Kumar
1BM22CS224



Lab - 9

20/2/24

(Q) Implementing event Handling in java.

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

class SwingDemo {

    SwingDemo() {

        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 180);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JLabel jlab = new JLabel("Enter the divider and
                    dividend:");

        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        JButton button = new JButton("Calculate");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
```

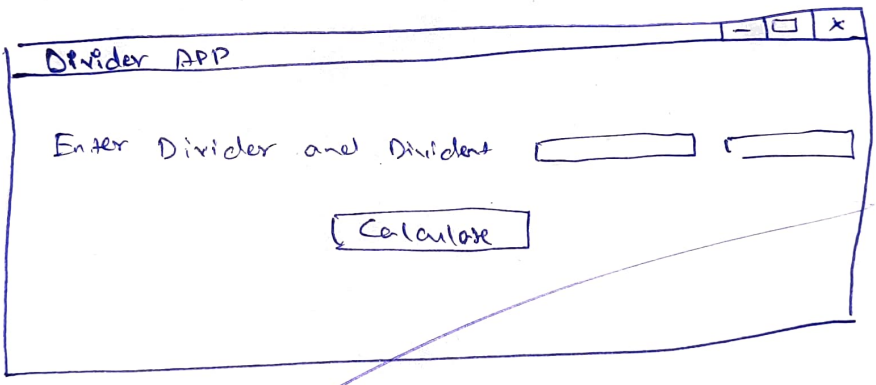
```
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(ansLab);

ActionListener l = new ActionListener() {
    public void actionPerformed(ActionEvent ev) {
        System.out.println("Action event from a text
            field");
    }
};

ajtf.addActionListener(l);
bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent ev) {
        try {
            int a = Integer.parseInt(ajtf.getText());
            int b = Integer.parseInt(bjtf.getText());
            if(b==0) {
                throw new ArithmeticException();
            }
            int ans = a/b;
```

```
alab.setText("\n A = " +a);
blab.setText("\n B= " +b);
ansLab.setText("\n Ans = " + ans);
err.setText("");
}
catch (NumberFormatException e) {
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Only Int");
}
catch (ArithmeticException e) {
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("Cant zero");
}
});
jfrm.setVisible(true);
}
```

```
public static void main (String [] args) {
    SwingUtilities.invokeLater (new Runnable () {
        public void run () {
            new SwingDemo();
        }
    });
}
}
```

Output :

Rishabh 1JUMAR, 1BM22CS221



20.12.24

LAB Program – 1

Q) To implement Quadratic Function using Java.

```
import java.util.Scanner;
class Quadratic
{
int a, b, c;
double r1, r2, d;
void getd()
{
Scanner s = new Scanner(System.in);
System.out.println(&quot;Enter the coefficients of a,b,c&quot;);
a = s.nextInt();
b = s.nextInt();
c = s.nextInt();
}
void compute()
{
while(a==0)
{
System.out.println(&quot;Not a quadratic equation&quot;);
System.out.println(&quot;Enter a non zero value for a:&quot;);
Scanner s = new Scanner(System.in);
a = s.nextInt();
}
d = b*b-4*a*c;
if(d==0)
{

r1 = (-b)/(2*a);
System.out.println(&quot;Roots are real and equal&quot;);
System.out.println(&quot;Roo1 = Root2 = &quot; + r1);
}
```

```
else if(d&gt;0)
{
r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);
System.out.println(&quot;Roots are real and distinct&quot;);
System.out.println(&quot;Roo1 = &quot; + r1 + &quot; Root2 = &quot; + r2);
}
else if(d&lt;0)
{
System.out.println(&quot;Roots are imaginary&quot;);
r1 = (-b)/(2*a);
r2 = Math.sqrt(-d)/(2*a);
System.out.println(&quot;Root1 = &quot; + r1 + &quot; + i&quot;+r2);
System.out.println(&quot;Root1 = &quot; + r1 + &quot; - i&quot;+r2);
}

}
}

class QuadraticMain
{
public static void main(String args[])
{
Quadratic q = new Quadratic();
q.getd();

q.compute();
}
}
```

Q) write a program to find the factorial of a given number

```
class factorial{
```

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

{

int fac=1;

System.out.println("Enter a number: ");

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();
for(int i=1; i<=n; i++){

fac=fac*i;

}

System.out.println("The factorial: "+fac);

}

}
```

Q) wap to find the given 5 digit int is a palindrome or not

```
class palindrome{

public static void main(String args[])

{

int n, t, rem, rev=0;

Scanner sc = new Scanner(System.in);
```

```
System.out.println("Enter a 5 digit number: ");

n = sc.nextInt();

t = n;
while(t>0){

rem = t%10;

rev = rev*10+rem;

t = t/10;

}

if(rev == n) {

System.out.println("Palindrome");

}

else {

System.out.println("not palindrome");

}

}

}
```

Q) WAP to check whether a number is prime or not

```
import java.util.*;
```

```
class isprime

{

static void isprime(int n)

{

int i,m=0,flag=0;

m=n/2;

if(n==0||n==1)

{

System.out.println(n+" is not a prime number.");

}

}

else

{

for(i=2;i<=m;i++)

{

if(n%i==0)

{

System.out.println(n+" is not a prime number.");
```

```
flag=1;

break;

}

if(flag==0)

{

System.out.println(n+" is a prime number.");

}

}

}

public static void main(String args[])

{

int i;

Scanner sc=new Scanner(System.in);

System.out.println("Enter the value of i: ");

i = sc.nextInt();

isprime(i);

}

}
```

Q) write a method (largest) to find the max of 3 nos

```
class largestmethod

{

static void largest(int i, int j, int k)

{

if(i>=j && i>=k){

System.out.println(i+" is the largest number");

}

if(j>=i && j>=k){

System.out.println(j+" is the largest number");

}

if(k>=i && k>=j){

System.out.println(k+" is the largest number");

}

}

public static void main(String args[])

{
```

```
int i,j,k;

i=Integer.parseInt(args[0]);

j=Integer.parseInt(args[1]);

k=Integer.parseInt(args[2]);

largest(i, j, k);

}

}
```

LAB-2 Program Source Code

(Q) A java program to calculate the SGPA of the student with the given conditions.

```
import java.util.Scanner;

class Student {
 private String usn;
 private String name;
 private int[] credits;
 private int[] marks;

 public Student() {
 usn = "";
 name = "";
 }

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

 System.out.print("Enter USN: ");
 usn = scanner.nextLine();

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

 System.out.print("Enter number of subjects: ");
```

```
int numSubjects = scanner.nextInt();

credits = new int[numSubjects];

marks = new int[numSubjects];


for (int i = 0; i < numSubjects; i++) {

System.out.print("Enter credits for subject " + (i + 1) + ": ");

credits[i] = scanner.nextInt();


System.out.print("Enter marks for subject " + (i + 1) + ": ");

marks[i] = scanner.nextInt();

}

}


public void displayDetails() {

System.out.println("USN: " + usn);

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

System.out.println("Subject-wise details:");

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

System.out.println("Subject " + (i + 1) + " - Credits: " + credits[i] + ", Marks: " + marks[i]);   }

}


public double calculateSGPA() {

double totalCredits = 0;

double totalGradePoints = 0;


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

totalCredits += credits[i];
```

```
 totalGradePoints += calculateGradePoints(marks[i]) * credits[i];
 }


 return totalGradePoints / totalCredits;   }


 private double calculateGradePoints(int marks) {   if
(marks >= 90) {
 return 10.0;
 } else if (marks >= 80) {
 return 9.0;
 } else if (marks >= 70) {
 return 8.0;
 } else if (marks >= 60) {
 return 7.0;
 } else if (marks >= 50) {
 return 6.0;
 } else if (marks >= 40) {
 return 5.0;
 } else {
 return 0.0;
 }
 }
}


 public class Main {
 public static void main(String[] args) {
Student student = new Student();
student.acceptDetails();
```

```
student.displayDetails();
System.out.println("SGPA: " + student.calculateSGPA());

}

}
```

LAB-3 Program Source Code

(Q) A java program to create a class Book and implement the given scenario.

```
class Book {

private String name;

private String author;

private double price;

private int numPages;


public Book(String name, String author, double price, int numPages){

this.name = name;

this.author = author;

this.price = price;

this.numPages = numPages;

}


public String getName(){

return name;
```

```
}
public void setName(String name){

this.name = name;

}


public String getAuthor(){

return author;

}


public void setAuthor(String author){

this.author = author;

}


public double getPrice(){

return price;

}


public void setPrice(double price){

this.price = price;

}


public int getNumPages(){

return numPages;

}


public void setNumPages(int numPages){

this.numPages = numPages;

}
```

```
public String toString() {
return "Book Details - Nane: " + name + ", Author: " + author + ", Price: $" + price + ", Number of Pages: " + numPages;

}

}



public class Main{

public static void main(String[] args){

int n=3;



Book[] books = new Book[n];



books[0] = new Book("The Catcher","J.D",15.99,224);

books[1] = new Book("To Kill","Lee",12.50,336);

books[2] = new Book("1984","Orwell",9.99,328);



System.out.println("Rishabh, 1BM22CS221");



for(int i =0;i<n;i++){

System.out.println(books[i].toString());

System.out.println();

}

}

}
```

LAB-4 Program Source Code

(Q) A java program to create a class Shape and implement the given scenario.

```
abstract class Shape{

protected int dim1;

protected int dim2;


public Shape(int dim1, int dim2){

this.dim1 = dim1;

this.dim2 = dim2;

}


public abstract void printArea();


}


class Rectangle extends Shape{

public Rectangle(int len, int wid){

super(len,wid);

}


public void printArea(){

int area = dim1 * dim2;

System.out.println("Area of Rectangle:" + area);

}

}


class Triangle extends Shape{

public Triangle(int base, int height){
```

```
super(base,height);

}
public void printArea(){

double area = 0.5 * dim1 * dim2;

System.out.println("Area of Triangle:" + area);

}



}

class Circle extends Shape{

public Circle(int rad){

super(rad,0);

}


public void printArea(){

double area = Math.PI * dim1 * dim1;

System.out.println("Area of Circle:" + area);

}

}


public class Main{

public static void main(String[] args){

Rectangle rectangle = new Rectangle(5,10);

Triangle triangle = new Triangle(4,6);  Circle

circle = new Circle(7);


rectangle.printArea();

triangle.printArea();

circle.printArea();
```

```
}
}
```

LAB-5 Program Source Code

(Q) A java program to create a class Bank and implement the given scenario.

```
import java.util.Scanner;

class Account{
String customerName;
int accountNumber;
String accountType;
double balance;

public Account(String customerName, int accountNumber, String accountType, double balance){
this.customerName = customerName;
this.accountNumber = accountNumber;
this.accountType = accountType;
this.balance = balance;
}

public void displayBalance(){
System.out.println("Account Balance: " + balance);
```

```
}

}
class SavingsAccount extends Account {

double interestRate;


public SavingsAccount(String customerName, int accountNumber, String accountType, double balance, double interestRate){

super(customerName,accountNumber, accountType, balance);

this.interestRate = interestRate;

}


public void computeInterest(){

balance += (balance * interestRate) / 100;

System.out.println("Interest rate added");

}


public void withdraw(double amount){

if (amount <= balance){

balance -= amount;

System.out.println("Withdrawal Successful");

}

else{

System.out.println("Insufficient Fund");

}

}

}



class CurrentAccount extends Account{
```

```
double minBalance;

double serviceCharge;
public CurrentAccount(String customerName, int accountNumber, String accountType, double balance ,double minBalance,
double serviceCharge){

super(customerName,accountNumber, accountType, balance);


this.minBalance = minBalance;

this.serviceCharge = serviceCharge;

}


public void withdraw(double amount){

if (balance - amount >= minBalance){

balance -= amount;

System.out.println("Withdrawal Successful");

}

else{

System.out.println("Insufficient Fund, Service charge will be applied");

balance -= serviceCharge;

}

}

}


public class Bank{

public static void main(String args[]){

SavingsAccount savings = new SavingsAccount("Jhon",1001,"Savings",5000,5);

CurrentAccount current = new CurrentAccount("Alice",2001,"Current",8000,1000,50);


savings.computeInterest();

savings.displayBalance();
```

```
savings.withdraw(2000);

savings.displayBalance();
current.displayBalance();

current.withdraw(5000);

current.displayBalance();

}

}
```

LAB-6 Program Source Code

(Q) A java program to create classes and packages, and implement the given scenario.

```
import CIE.Internals;

import SEE.External;


public class CalculateFinalMarks{


 public static void  main(String[] args) {

 Internals student1 = Internals("123","John Doe", 5, new int[]{85,90,78,92,88});   External student1

= External("189","Rock", 5, new int[]{75,60,98,92,68});


 int finalMarks1 = calculateFinalMarks(student1.internalMarks);

 int finalMarks2 = calculateFinalMarks(student.seeMarks);


 System.out.println("Student 1 - Final Marks: " + finalMarks1);

 System.out.println("Student 2 - Final Marks: " + finalMarks2);

}
```

```
 private static int calculateFinalMarks(int[] marks) {
 int totalMarks = 0;

 for (int mark : marks) {

 totalMarks += mark;

 }

 return totalMarks;

 }

}
```

```
//Student.java

package CIE;


public class Student {

 String usn, name;

 int sem;


 public Student(String usn, String name, int sem) {

this.usn = usn;

 this.name = name;

 this.sem = sem;

 }

}


// CIE/Internals.java

package CIE;


public class Internals extends Student {
```

```
int[] internalMarks;
public Internals(String usn, String name, int sem, int[] internalMarks) {

super(usn, name, sem);

this.internalMarks = internalMarks;

}

}
```

```
//External.java

package SEE;


import CIE.Student;


public class External extends Student {

int[] seeMarks;


public External(String usn, String name, int sem, int[] seeMarks) {

super(usn, name, sem);

this.seeMarks = seeMarks;

}

}
```

LAB-7 Program Source Code

(Q) A java program to create class Father and inherit Son, and implement the given scenario.

```
import java.util.Scanner;


class WrongAge extends Exception {
 WrongAge(String message) {
 super(message);
 }
}

class InputScanner {
 static Scanner scanner = new Scanner(System.in);  }

class Father extends InputScanner {
 int fatherAge;

 Father() throws WrongAge {
 System.out.print("Enter Father's age: ");
fatherAge = scanner.nextInt();

 if (fatherAge < 0) {
 throw new WrongAge("Age cannot be negative");   }
 }

 void display() {
```

```
System.out.println("Father's Age: " + fatherAge);   }

}


class Son extends Father {
 int sonAge;


 Son() throws WrongAge {

 System.out.print("Enter Son's age: ");

 sonAge = scanner.nextInt();


 if (sonAge > fatherAge) {

 throw new WrongAge("Son's age cannot be greater than father's age");   } else if

(sonAge < 0) {

 throw new WrongAge("Age cannot be negative");   }

 }


 void display() {

 super.display();

 System.out.println("Son's Age: " + sonAge);

 }

}


public class ExceptionHandlingDemo {

 public static void main(String[] args) {

 try {

 Son son = new Son();

 son.display();

 } catch (WrongAge e) {
```

```
System.out.println("Exception: " + e.getMessage());   }
}
}
```

LAB-8 Program Source Code

(Q) A java program to create two threads and implement the given scenario.

```
class DisplayThread extends Thread{
 private String message;
 private int sleepTime;

 public DisplayThread(String message, int sleepTime){
 this.message = message;
 this.sleepTime = sleepTime;
 }

  //@override
 public void run(){
 while(true){
 System.out.println(message);
 try{
 Thread.sleep(sleepTime * 1000);
 }
 catch(InterruptedException e){
 e.printStackTrace();
 }
 }
```

```
}

}
class Main{

public static void main(String [] args){

DisplayThread thread1 = new DisplayThread("BMSCE", 10);

DisplayThread thread2 = new DisplayThread("CSE", 2);

thread1.start();

thread2.start();

}

}
```

LAB-9 Program Source Code

(Q) A java program to create an Event Handling Function and implement the given scenario.

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;


class SwingDemo {

SwingDemo() {

// create jframe container

JFrame jfrm = new JFrame("Divider App");

jfrm.setSize(275, 150);

jfrm.setLayout(new FlowLayout());

// to terminate on close
```

```
jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// text label

JLabel jlab = new JLabel("Enter the divider and dividend:");



// add text field for both numbers

JTextField ajtf = new JTextField(8);

JTextField bjtf = new JTextField(8);



// calc button

JButton button = new JButton("Calculate");



// labels

JLabel err = new JLabel();

JLabel alab = new JLabel();

JLabel blab = new JLabel();

JLabel anslab = new JLabel();



// add in order :)

jfrm.add(err); // to display error

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);



ActionListener l = new ActionListener() {
```

```
 public void actionPerformed(ActionEvent evt) {
System.out.println("Action event from a text field");
 }
 };
 ajtf.addActionListener(l);
 bjtf.addActionListener(l);

 button.addActionListener(new ActionListener() {   public void
actionPerformed(ActionEvent evt) {   try {
    int    a    =    Integer.parseInt(ajtf.getText());            int    b    =
Integer.parseInt(bjtf.getText());   if (b == 0) {
 throw new ArithmeticException();   }
 int ans = a / b;

 alab.setText("\nA = " + a);   blab.setText("\nB = " + b);
anslab.setText("\nAns = " + ans);   err.setText("");
 } catch (NumberFormatException e) {   alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("Enter Only Integers!");   } catch
(ArithmeticException e) {   alab.setText("");
 blab.setText("");
 anslab.setText("");
 err.setText("B should be non-zero!");
 }
 }
 });
```

```
// display frame
jfrm.setVisible(true);
}

public static void main(String args[]) {
// create frame on event dispatching thread
SwingUtilities.invokeLater(new Runnable() {
public void run() {
new SwingDemo();
}
});
}
}
```

LAB-10 Program Source Code

(Q) A java program to create an IPR and implement the given scenario.

```
public class DeadlockExample {

public static void main(String[] args) {
final SharedResource sharedResource = new SharedResource();
Thread process1 = new Thread(() -> {
try {
```

```
    sharedResource.method1();             }     catch
(InterruptedException e) {   e.printStackTrace();
 }
 });

 Thread process2 = new Thread(() -> {
 try {
    sharedResource.method2();             }     catch
(InterruptedException e) {   e.printStackTrace();
 }
 });

 process1.start();
 process2.start();
 }
}

class SharedResource {
 private final Object lock1 = new Object();
 private final Object lock2 = new Object();

 public void method1() throws InterruptedException {
synchronized (lock1) {
 System.out.println("Method 1 acquired lock1");
 Thread.sleep(1000);

 synchronized (lock2) {
 System.out.println("Method 1 acquired lock2");   // Perform some
```

```
task using both lock1 and lock2   }
 }
 }

 public void method2() throws InterruptedException {
synchronized (lock2) {
 System.out.println("Method 2 acquired lock2");
Thread.sleep(1000);

 synchronized (lock1) {
 System.out.println("Method 2 acquired lock1");   // Perform some
task using both lock1 and lock2   }
 }
 }
}
```