Lab prog 1:

Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b,

c and use the quadratic formula. If the discriminant b 2 -4ac is negative, display a message stating that

there are no real solutions

```
import java.util.*;

class quadratic
{
	int a,b,c;
	double r1,r2,d;
	void getd()
	{
		Scanner s=new Scanner(System.in);
		System.out.println("enter the coeffients 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 for 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 an equal");
			System.out.println("root1=root2="+r1);
		}
```

```
                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 QuadraticMain
{
        public static void main(String agrs[])
        {
                quadratic q=new quadratic();
                q.getd();
                q.compute();
        }
}
```

Lab prog 2:

Develop a Java program to create a class Student with members usn, name, an array credits and an

array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
//Lab program 2

import java.util.*;
class Subject
{
        int subjectMarks;
        int credits;
        int grade;
}
class Student
```

```
{
	Subject subject[];
	String name;
	String usn;
	double sgpa;
	Scanner s;
	Student()
	{
		int i;
		subject =new Subject[9];
		for(i=0;i<9;i++)
			subject[i]=new Subject();
		s=new Scanner(System.in);
	}
	void getStudentDetails()
	{
		System.out.println("Enter student name");
		name=s.next();
		System.out.println("Enter student USN");
		usn=s.next();
	}
	void getMarks()
	{
		for(int i=0;i<9;i++)
		{
			System.out.println("Enter marks");
			subject[i].subjectMarks=s.nextInt();
			System.out.println("Enter number of +credits");
			subject[i].credits=s.nextInt();
			subject[i].grade=(subject[i].subjectMarks/10)+1;
			if(subject[i].grade==11)
				subject[i].grade=10;
			if(subject[i].grade<=4)
				subject[i].grade=0;

		}
	}
	void computeSGPA()
	{
		int effectiveScores=0;
		int totalCredits=0;
		for(int i=0;i<9;i++)
		{
			effectiveScores+=subject[i].grade*subject[i].credits;
```

```
                totalCredits+=subject[i].credits;
            }
            sgpa=(double)effectiveScores/(double)totalCredits;
        }
}


class Main
{
        public static void main(String args[])
        {
            Student s1=new Student();
            s1.getStudentDetails();
            s1.getMarks();
            s1.computeSGPA();
            System.out.println("Student name:"+s1.name);
            System.out.println("Student usn:"+s1.usn);
            System.out.println("Student SGPA:"+s1.sgpa);
        }
}
```

Lab prog 3:

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

```
import java.util.Scanner;
```

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

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

    public String toString() {
        return "Book name: " + name + "\n" +
                "Author name: " + author + "\n" +
                "Price: " + price + "\n" +
                "Number of pages: " + numPages + "\n";
    }
}

class BooksMain {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        int n;

        System.out.println("Enter the number of books: ");
        n = s.nextInt();
        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter the name of the book");
            String name = s.next();
            System.out.println("Enter the author of the book");
            String author = s.next();
            System.out.println("Enter the price of the book");
            int price = s.nextInt();
            System.out.println("Enter the pages of the book");
            int numPages = s.nextInt();
            books[i] = new Book(name, author, price, numPages);
        }

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

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

Lab prog 4:

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner;
abstract class shape
{
        double dim1, dim2, radius;
        shape(double a,double b)
        {
                dim1=a;
                dim2=b;
        }
        shape(double a)
        {
                radius=a;
        }
        abstract void area();
}

class rectangle extends shape
```

```
{
	rectangle(double a, double b)
	{
		super(a,b);
	}
	void area()
	{
		System.out.println("area of rectangle is : "+(dim1*dim2));
	}
}

class triangle extends shape
{
	triangle (double a,double b)
	{
		super(a,b);
	}
	void area()
	{
		System.out.println("area of triangle is : "+(dim1*dim2)/2);
	}
}

class circle extends shape
{
	circle(double a)
	{
		super(a);
	}
	void area()
	{
		System.out.println("area of circle is : "+(3.14*(radius)*(radius)));
	}
}

class AbstractAreaMain
{
	public static void main(String args[])
	{
		Scanner s=new Scanner(System.in);
		System.out.println("enter your name : sanvi N");
		System.out.println("enter your usn : 1BM22CS245 ");

		System.out.println("enter length and breadth of rectangle: ");
```

```
                double l=s.nextInt();
                double b=s.nextInt();
                System.out.println("enter base and height of triangle: ");
                double h1=s.nextInt();
                double h2=s.nextInt();
                System.out.println("enter the circle: ");
                double r=s.nextInt();
                shape sh;
                rectangle rect=new rectangle(l,b);
                triangle tri=new triangle(h1,h2);
                circle cir=new circle(r);
                sh=rect;
                sh.area();
                sh=tri;
                sh.area();
                sh=cir;
                sh.area();
        }
}
```

Lab prog 5:

Develop a Java program to create a class Bank that maintains two kinds of account for its

customers, one called savings account and the other current account. The savings account

provides compound interest and withdrawal facilities but no cheque book facility. The

current account provides cheque book facility but no interest. Current account holders

should also maintain a minimum balance and if the balance falls below this level, a service

charge is imposed.

```
import java.util.Scanner;
class account
{
	String name;
	int accno;
	String type;
	double balance;

	account(String name,int accno,String type,double balance)
	{
		this.name=name;
		this.accno=accno;
		this.type=type;
		this.balance=balance;
	}
	void deposit(double amount)
	{
		balance+=amount;
	}
	void withdraw(double amount)
	{
		if((balance-amount)>=0)
		{
			balance-=amount;
		}
		else
		{
			System.out.println("insufficient balance,can't withdraw");
		}
	}

	void display()
	{

System.out.println("name:"+name+"accno:"+accno+"type:"+type+"balance:"+balance);
	}
}
class savAcct extends account
{

	private static double rate=5;
	savAcct(String name,int accno,double balance)
	{
		super(name,accno,"savings",balance);
```

```
	}

	void interest()
	{
		balance+=balance*(rate)/100;
		System.out.println("balance:"+balance);
	}


}
class curAcct extends account
{

	private double minBal=500;
	private double serviceCharges=50;

	curAcct(String name,int accno,double balance)
	{
		super(name,accno,"current",balance);

	}


	void checkmin()
	{

		if(balance<minBal)
		{
			System.out.println("balance is less than min balance,service charges
				imposed:"+serviceCharges);
			balance-=serviceCharges;
			System.out.println("balance is:"+balance);
		}

	}

}
class accountMain
{
	public static void main(String a[])
	{
		Scanner s=new Scanner(System.in);
		System.out.println("enter the name :");
```

```
String name=s.next();
System.out.println("enter the type(current/savings):");
String type=s.next();
System.out.println("enter the account number:");
int accno=s.nextInt();
System.out.println("enter the intial balance:");
double balance=s.nextDouble();
int ch;
double amount1,amount2;
account acc=new account(name,accno,type,balance);
savAcct sa=new savAcct(name,accno,balance);
curAcct ca=new curAcct(name,accno,balance);
while(true)
{
        if(acc.type.equals("savings"))
        {
                System.out.println("\nMenu\n1.deposit 2.withdraw 3.compute
interest                                4.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
                switch(ch)
                {
                        case 1:System.out.println("enter the amount:");
                        amount1=s.nextInt();
                        sa.deposit(amount1);
                        break;
                        case 2:System.out.println("enter the amount:");
                        amount2=s.nextInt();
                        sa.withdraw(amount2);
                        break;
                        case 3:sa.interest();
                        break;
                        case 4:sa.display();
                        break;
                        case 5:System.exit(0);
                        default:System.out.println("invalid input");
                        break;
                }
        }
        else
        {
                System.out.println("\nMenu\n1.deposit 2.withdraw  3.display");
                System.out.println("enter the choice:");
                ch=s.nextInt();
```

```
                        switch(ch)
                        {
                                case 1:System.out.println("enter the amount:");
                                amount1=s.nextInt();
                                ca.deposit(amount1);
                                break;
                                case 2:System.out.println("enter the amount:");
                                amount2=s.nextInt();
                                ca.withdraw(amount2);
                                ca.checkmin();
                                break;

                                case 3:ca.display();
                                break;
                                case 4:System.exit(0);
                                default:System.out.println("invalid input");
                                break;
                        }
                }
        }




    }
}
```

Lab prog 6:

Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
//Student.java
package CIE;

import java.util.Scanner;

public class Student {

protected String usn = new String();
protected String name = new String();
protected int sem;

public void inputStudentDetails()
{
Scanner sc=new Scanner(System.in);
System.out.println(&amp;quot;Enter student usn&amp;quot;);
usn=sc.next();
System.out.println(&amp;quot;Enter student name&amp;quot;);
name=sc.next();
System.out.println(&amp;quot;Enter student semester&amp;quot;);
sem=sc.nextInt();
}

public void displayStudentDetails() {
```

```
System.out.println(&quot;student usn:&quot;+ usn);
System.out.println(&quot;student name:&quot;+n ame);
System.out.println(&quot; student sem:&quot;+ sem);
}
}

//Internals.java
package CIE;
import java.util.Scanner;

public class Internals extends Student {

protected int marks[] = new int[5];

public void inputCIEmarks()
{
Scanner sc=new Scanner(System.in);
for (int i=0;i&lt;5;i++)
{
System.out.println(&quot;Enter 5 subject marks&quot;);
marks[i]=sc.nextInt();
}
}
}

//Externals.java
package SEE;

import CIE.Internals;

import java.util.Scanner;

public class Externals extends Internals {

protected int marks[];

protected int finalMarks[];

public Externals() {

marks = new int[5];
finalMarks = new int[5];
}
```

```
public void inputSEEmarks()
{

Scanner sc = new Scanner(System.in);

for(int i=0;i&amp;lt;5;i++)
{

System.out.print(&amp;quot;Subject &amp;quot;+(i+1)+&amp;quot; marks: &amp;quot;);

marks[i] = sc.nextInt();
}
}

public void calculateFinalMarks() {

for(int i=0;i&amp;lt;5;i++)

finalMarks[i] = marks[i]/2 + super.marks[i];

}

public void displayFinalMarks() {

displayStudentDetails();

for(int i=0;i&amp;lt;5;i++)

System.out.println(&amp;quot;Subject &amp;quot; + (i+1) + &amp;quot;: &amp;quot; + finalMarks[i]);

}

}

//Main1.java
import SEE.Externals;

class Main1 {

public static void main(String args[])

{

int numOfStudents = 2;
```

```
Externals finalMarks[] = new
Externals[numOfStudents];

for(int i=0;i&amp;lt;numOfStudents;i++)

{

finalMarks[i] = new Externals();
finalMarks[i].inputStudentDetails();

System.out.println(&amp;quot;Enter CIE marks&amp;quot;);

finalMarks[i].inputCIEmarks();

System.out.println(&amp;quot;Enter SEE marks&amp;quot;);

finalMarks[i].inputSEEmarks();

}
System.out.println(&amp;quot;Displaying data:\n&amp;quot;);

for(int i=0;i&amp;lt;numOfStudents;i++)

{

finalMarks[i].calculateFinalMarks();

finalMarks[i].displayFinalMarks();

} //end of for loop

} // end of public main

} //end of class main
```

Lab prog 7:

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called

“Father” and derived class called “Son” which extends the base class. In Father class, implement a

constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class,

implement a constructor that cases both father and son’s age and throws an exception if son’s age is

>=father’s age.

```
import java.util.*;

// Custom exception class
class WrongAge extends Exception {
    public WrongAge() {
        super("Age Error");
    }

    public WrongAge(String message) {
        super(message);
    }
}

// Input scanner class
class InputScanner {
    protected Scanner scanner;

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

    public int nextInt() {
        return scanner.nextInt();
```

```
    }
}

// Father class extending InputScanner
class Father extends InputScanner {
    public int fatherAge;

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

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

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

// Son class extending Father
class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
        super(); // Call the constructor of the base class (Father)

        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");
        }
    }

    @Override
    public void display() {
        super.display(); // Call the display method of the base class (Father)
        System.out.println("Son's age: " + sonAge);
    }
}
```

```
public class FamMain {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

Lab prog 8:

Write a program which creates two threads, one

thread displaying “BMS College of Engineering” once

every ten seconds and another displaying “CSE” once

every two seconds.

```
class bms extends Thread{
        public void run(){
                for(int i=1;i<=5;i++){
                        System.out.println("bms college of engineering"+i);
                        try
                        {
                                Thread.sleep(10000);
                        }
                        catch(InterruptedException e){
                                e.printStackTrace();
                        }
                        }
                }
}
class cse extends Thread{
        public void run(){
                for(int i=1;i<=10;i++){
```

```
                        System.out.println("cse"+i);
                        try
                        {
                                Thread.sleep(2000);
                        }
                        catch(InterruptedException e){
                                e.printStackTrace();
                        }
                        }
                }
}
class threadMain{
        public static void main(String a[]){
                bms obj1=new bms();
                cse obj2=new cse();
                obj1.start();
                obj2.start();
        }
}
```

Lab prog 9:

Write a program that creates a user interface to perform integer divisions.

The user enters two numbers in the text fields, Num1 and Num2. The division of

Num1 and Num2 are displayed in the Result field when the Divide button is clicked.

If Num1 or Num2 were not an integer, the program would throw a

NumberFormatException. If Num2 were Zero, the program would throw an

Arithmetic Exception Display the exception in a message dialog box.

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

public class SwingDemo {

    public 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 divident:");

// 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 bois
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

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("Division by zero is not allowed.");
            }
            int ans = a / b;
```

```
                alab.setText("\nA = " + a);
                blab.setText("\nB = " + b);
                anslab.setText("\nAns = " + ans);

            } 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 prog 10:

Demonstrate Inter process Communication and deadlock

A-

PCFixed-

```
class Q {
	int n;
	boolean valueSet = false;
	synchronized int get() {
		while(!valueSet){
			try {
				System.out.println("Consumer waiting");
				wait();
			} catch(InterruptedException e) {
			System.out.println("InterruptedException caught");
			}

			}
		System.out.println("Got: " + n);
		valueSet = false;
		System.out.println("Intimate Producer\n");
		notify();
		return n;
	}
	synchronized void put(int n) {
		while(valueSet){
			try {
				System.out.println("Producer waiting");
				wait();
				} catch(InterruptedException e) {
					System.out.println("InterruptedException caught");
				}

			}
		this.n = n;
```

```
                valueSet = true;
                System.out.println("Put: " + n);
                System.out.println("Intimate Consumer\n");
                notify();
                }

}

class Producer implements Runnable {
        Q q;
        Producer(Q q) {
                this.q = q;
                new Thread(this, "Producer").start();
        }
        public void run() {
        int i = 0;
        while(i<5) {
                q.put(i++);
        }
        }
}

class Consumer implements Runnable {
        Q q;
        Consumer(Q q) {
                this.q = q;
                new Thread(this, "Consumer").start();
        }
        public void run() {
                int i=0;
                while(i<5) {
                        int r=q.get();
                        System.out.println("consumed:"+r);
                        i++;
                }
        }
}
class PCFixed {
        public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
        }
```

```
}
```

B-

Deadlock:

```
class A {

    synchronized void foo(B b) {

        String name = Thread.currentThread().getName();

        System.out.println(name + " entered A.foo");

        try {

            Thread.sleep(1000);

        } catch (Exception e) {

            System.out.println("A Interrupted");

        }
        System.out.println(name + " trying to call B.last()");
        b.last(); // this line will be executed after B.last() is executed

        System.out.println(name + " finished executing A.foo");

    }

    synchronized void last() {

        System.out.println("Inside A.last");

    }

}
```

```
class B {

    synchronized void bar(A a) {

        String name = Thread.currentThread().getName();

        System.out.println(name + " entered B.bar");

        try {

            Thread.sleep(1000);

        } catch (Exception e) {

            System.out.println("B Interrupted");

        }
        System.out.println(name + " trying to call A.last()");
        a.last(); // this line will be executed after A.last() is executed

        System.out.println(name + " finished executing B.bar");

    }

    synchronized void last() {

        System.out.println("Inside B.last");

    }

}

class Deadlock implements Runnable {

    A a = new A();

    B b = new B();

    Deadlock() {

        Thread.currentThread().setName("MainThread");

        Thread t = new Thread(this, "RacingThread");
```

```
        t.start();
        a.foo(b); // get lock on a in this thread.
        System.out.println("Back in mainthread");
    }
    public void run() {
        b.bar(a); // get lock on b in other thread.
        System.out.println("Back in other thread");
    }
    public static void main(String args[]) {
        new Deadlock();
    }
}
```

INDEX

NAME: Sanvi Nadiga 1BM22CS245 SUBJECT: Java Lab

STD.: DIV.: ROLL NO.: SCHOOL:

S.No.	Date	Title	Page No.	Teacher's Sign/Remarks
1.	5/12/23	Sample Prog.		
2.	12/12/23	Program 1 - Quadratic Eq.	10	8 12/12/23
3.	19/12/23	Prog 2 - Student SGPA	10	8 19/12/23
4.	26/12/23	Prog 3 - Books Prog.	10	8 26/12/23
5.	2/1/24	Prog 4 - Abstract Area.	10	8 2/1/2024
6.	16/1/24	Prog 5 - Bank account.	10	8 16/1/2024
7.	16/1/24	Prog 6 - Strings (additional Prog)		
8.	23/1/24	Prog 7 - Packages	10	8 30/1/2024
9.	30/1/24	Prog 8 - Try & catch	10	8 30/1/2024
10.	6/2/24	Prog 9 - MultiThreads	10	8
11.	6/2/24	Prog 10 - Inter Process Communication & deadlock.	10	8 6/2/20

S.No.	Date	Title	Page No.	Teacher's Sign/Remark
12.	13/2/24	Deadlock (Lab 11) Prog 10.b	13/2/2024	
13.	20/2/24	Applet Prog in Java Lab 9.	20/2/2024	

Quick Work

Page No.:

Date:

M T W T F S S

Program to say hello world

```
import java.util.*;
public class demo1{
    public static void main(String args[])
{
    System.out.println("hello world");
}
}
```

O/p: Hello world

Program to add 2 numbers

```
import java.util.*;
class add{
  public static void main(String args[])
{
    int a=10;
    int b=15;
    int sum;
    sum = a+b;
    System.out.println("sum = "+sum);
}
}
```

O/p:

15

Program to multiply 2 numbers

```
import java.util.*;
class multiply{
public static void main(String args[]
{
    int a=5;
    int b=25;
    int sum= a*b;
    System.out.println("sum=" +sum);
}
}
```

Op: 125

Program to compare 2 numbers

```
import java.util.*;
class comp{
public static void main(String args[])
{
    int a=10;
    int b=25;
    if(a>b)
    {
    system.out.println("A is greater");
    }
```

Quick Work
Page No.:
Date:
M T W T F S S

```
else if (a<b)
{
    System.out.println("B is greater")
}else
{
    System.Out.println("Equal");
}
}
```

O/P:

B is greater

12/12/23

1. Develop a java program that prints all real solutions to the quadratic equation $ax^2 + bx + c = 0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2 - 4ac$ is -ve, display a message stating that there are no real sols.

```
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
            coefficients 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 for 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
                read and equal");
        System.out.println("Root1 = Root2 =
                "+r1);
    }
    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 QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
```

```
of: getd();
a: compute();
}
```

O/p:

3 1 2
roots

Sanvi Nadiga
1BM22CS245

1. enter the coefficients of a, b, c
3 1 2
roots are imaginary
root 1 = 0.0 + i0.7993052538854537
root 2 = 0.0 - i0.7993052538854537

2. enter the coefficients of a, b, c
1 4 1
root 1 = -2.0 + i NAN
root 2 = -2.0 - i NAN

3. enter the coefficients of a, b, c
1 2 1
roots are real & equal
root 1 = root 2 = -1.0

12/12/2023

Quick Work

Page No.:

Date:

M T W T F S S

2. To write a program in Java to find the area of a rectangle and verify the same with various inputs (length, breadth).

```
~~class~~
~~java.util~~
import java.util.*;
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("length of
        rectangle = " + length);
    System.out.println("breadth of
                rectangle = " + breadth);

    System.out.println("area of
        rectangle = " + area);

}

}
```

Sanvi Nadiga
1BM22CS245

2. import java.util.*;
class subject {
int Subject_marks;
int credits;
int grade;
}

class student
{
subject subject[];
String name;
String USN;
double SGPA;
Scanner s;
student()
{
int i;
Subject = new Subject[
for(i=0;i<a;i++)
{ subject[i]=new
Subject
s=new Scanner(S

}

void getStudentDetails()
{
System.out.println("enter Student
name");
name = s.next()
System.out.println("enter Stud

Quick Work
Page No.:
Date:
M T W T F S S

```
        USN = s.next();
    }

    void getMarks()
    {
        for(int i=0; i<9; i++)

            System.out.println("Enter subject
                marks");
            subject[i].subject_marks = s.next
                Int();

            System.out.println("Enter the
                subject credits");
            subject[i].credits = s.nextInt();
            subject[i].grade = subject[i].
                subject_marks/100) + 1;

            if(subject[i].grade == 10;
            {
            if(  subject[i].grade = 10;
            }
            if(subject[i].grad >= 4)
            {
                subject[i].grade = 0;
            }
        }
    }

    void computeSGPA()
    {
        int effective_score = 0;
```

```
int totalScore=0;
for(int i=0; i<9; i++)
{
    effectiveScore+=(subject[i].
        grade * subject[i].credits);
    totalCredits+=subject[i].credits;
};
SGPA=(double)effectiveScore/
    (double)totalCredits;
};
};
class Main1
{
    public static void main(String args[])
    {
        Student s1=new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("name: "+s1.name);
        System.out.println("USN: "+s1.USN);
        System.out.println("SGPA: "+s1.SGPA);
    }
}
```

Quick Work

Page No.:

Date:

M T W T F S S

O/p:

Enter Student Name

Sawi

Enter Student USN

1BM22CS245

Enter Marks

98

Enter no. of credits

4

Enter marks

89

Enter credits

3

Enter marks

91

Enter credits

3

Enter marks

87

Enter credits

3

Enter marks

78

Enter credits

2

Enter marks

87

Enter credits

3

Enter marks

93

Sawi Nadiga
1BM22CS245

Enter credits
3

Enter marks
89
Enter credits
3
Enter marks
86
Enter credits
3

student name : sanvi
student USN = 1BM22CS268
student SGPA : 9.18818

19/12/23

26/12/23

Quick Work
Page No.:
Date:
M T W T F S S

Lab Prog - 3

Q Create a class : Book which contains 4 members; name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a java program to create n book objects.

```
import java.util.Scanner;
class Book{
    String name;
    String author;
    int price;
    int numPages;

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

public String toString()
{
    return "Book name: " + name + "\n" +
        "Author name: " + author + "\n"
```

```
+ "Price: " + price + "\n" +
"Number of pages: " + numPages
+ "\n";
    }
}

Class BooksMain {
    public static void main (String args[]) {
        Scanner s = new Scanner (System.in);
        int n;
        System.out.println("Enter the number
        of books : ");
        n = s.nextInt();
        Books[] books = new book[n];

        for(int i = 0; i<n; i++)
        {
        System.out.println("Enter the
        name the book");
        String name = s.next();
        System.out.println("Enter the
        author of the book");
        String author = s.next();
        System.out.println("Enter the
        price of the book");
        int price = s.nextInt();
        System.out.println("Enter the
        pages of the book");
        int numPages = s.nextInt();
```

```
        book[i] = new Book (name, author, price,
                            numpages);
    }
    for (int i = 0; i < n; i++)
    {
        System.out.println(book[i].
                           toString());
    }
}
```

Output :

Enter no. of books :
2
Enter the name of the book
wow
Enter the author of the book
owo
Enter the price of the book
24
Enter the pages of the book
234
Enter the name of the book
idk
Enter the author of the book
Suresh
Enter the price of the book
34
Enter the pages of the book

321

Book name : wow
Author name : owo
Price : 24
Number of pages : 234

Book name : idk
Author name : Suresh
Price : 32
Number of pages : 321

26/12/23

- Sanvi Nadiga
1BM22CS245

2/9/24

Quick Work
Page No.:
Date:
M T W T F S S

Q. Develop a Java program to create an abstract class named Shape that contains 2 integers and an empty method name printArea(). Provide 3 classes named Rectangle, Triangle and circle such that each of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

Steps:

1. Add class InputScanner
2. Add class Shape extend InputScanner
3. Add class Rectangle extends Shape.
4. Add class Triangle extends Shape
5. Add class circle extends shape.

// Add

⇒ import java.util.Scanner;
abstract class shape
{
    double dim1, dim2, radius;
    shape (double a, double b)
    {
        dim1 = a
        dim2 = b
    }
    shape (double a)
    {

```
        radius = a;
    }
    abstract void area();
}
class rectangle extends shape
{
    rectangle (double a, double b)
    {
        super(a,b);
    }
    void area()
    {
        System.out.println("area of rect is:
            + (dim1 * dim2));
    }
}

class triangle extends shape
{
    triangle (double a, double b)
    {
        super (a,b);
    }
    void area()
    {
        System.out.println("area of triangle is
            + (dim1 * dim2)/2);
    }
}
```

Quick Work

Page No.:

Date:

M T W T F S S

class circle extends shape

{

circle(double a)

{

super(a);

}

void area()

{

System.out.println("area of circle is:
" +(3.14*(radius)*radius));

}

}

class AbstractAreaMain

{

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

System.out.println("enter name:
");

System.out.println("enter your usn
1BM22CS245");

System.out.println("enter length &
breadth of rect:");

double l = s.nextInt();

double b = s.nextInt();

System.out.println("enter base and
height of triangle:");

double r = s.nextInt();

shape sh;

```
rectangle rect = new rectangle(l, b);
triangle tri = new triangle(ht, ht2);
circle cir = new circle(ri);
sh = rect;
sh.area();
sh = tri;
sh.area();
sh = cir;
sh.area();
}
```

O/P:

```
enter length and breadth of rect:
5 10
enter base and height of triangle:
5 10
enter the circle:
25
area of rect is : 50.0
area of triangle is : 25.0
area of circle is : 1962.5
```

2/1/2024

Samir Nadiga
1BM22CS245

Quick Work
Page No.:
Date:
M T W T F S S

LAB-5

Develop a Java program to create a class Bank that maintains 2 kinds of account for its customers, one called savings account and the other current account. The savings account current account provides cheque book facility but no interest. Current account holders should also maintain a min balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customers name, account number & type of account. From this derive the classes cur-acct and sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.
b) Display the balance.
c) Compute and deposit interest
d) Permit withdrawal and update the balance.

Check for the min. balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class account
{
    String name;
    int accno;
    String type;
    double balance;
    account(String name, int accno, String type,
            double balance)
    {
        this.name = name;
        this.accno = accno;
        this.type = type;
        this.balance = balance;
    }
    void deposit(double amount)
    {
        balance += amount;
    }
    void withdraw(double amount)
    {
        if((balance - amount) >= 0)
        {
            balance -= amount;
        }
        else
        {
            System.out.println("
            insufficient balance; can't
            withdraw");
        }
```

Quick Work
Page No.:
Date:
M T W T F S S

```
}
void display()
{
    System.out.println("name: " + name
    "accno: " + accno + "type: " +
    type + "balance: " + balance);
}

}

class savAcct extends account
{
    private static double rate = 5;
    savAcct(String name, int accno, double
    balance)
    {
        super(name, accno, "savings",
            balance);
    }
    void interest()
    {
        balance += balance * (rate)/100;
        System.out.println("balance:
            + balance
    }

}
class curAcct extends account
{
```

Quick Work
Page No.:
Date:
M T W T F S S

```
private double minbal=500;
Private double ServiceCharges=50;

curAcct(String name, int accno,
    double balance)
{
    Super(name, accno, "current",
    balance);

}

void checkmin()
{
    if(balance<minBal)
    {
        System.out.println("balance
        is less than min balance,
        service charges imposed:"
        +ServiceCharges);
        balance-=ServiceCharges;
        System.out.println("balance is:"
        +balance);
    }
}

class accountMain
{
    public static void main(String a[])
    {
        Scanner s=new Scanner(System.in
```

Quick Work
Page No.:
Date:
M T W T F S S

System
System.out.println("enter the name:");
String name = s.next();
System.out.println("enter the type (current / savings):");
String type = s.next();
System.out.println("enter the account number:");
int accno = s.nextInt();
System.out.println("Enter the initial balance:");
double balance = s.nextDouble();
int ch;
double amount1, amount2;
account acc = new account(name, accno, type, balance);
SavAct sa = new SavAct(name, accno, balance);
crAct ca = new crAct(name, accno, balance);

while (true)
{
if (a.cc.type.equals("savings"))
{
System.out.println("1. deposit 2. withdraw 3. compute interest 4. display");
System.out.println("

```
Enter the amount: ");
amount2 = s.nextInt();
ca.withdraw(amount2);
break;
case 2: System.out.println
("enter the amount:");
amount2 = s.nextInt();
ca.withdraw(amount2);
break;
case 3: ca.interest();
break;
case 4: ca.display();
break;
case 5: System.exit(0);
default: System.out.println
("invalid input");
break;
}
}
else
{
System.out.println("1.new/4.
deposit 2.withdraw 3.display");
system.out.println("enter the
choice:");
ch=s.nextInt();
switch(ch)
{
case 1: System.out.println
("enter the amount:");
amount1 = s.nextInt();
ca.deposit(amount1);
```

```
        break;
        case 2 : System.out.println("
          enter the amount : ");
        amount2 = sinent Int();
        ca.withdraw(amount2);
        ca.checkmin();
        break;

        case 3: ca.display();
        break;
        case 4: System.exit(0);
        default : System.out.println
          ("Invalid input");
        break;

    }
  }
}
```

Output :

enter your name:
Sam:
enter the account number:
300
enter the initial balance:
5000

menu
1. display 2. withdraw 3. compute Interest
enter the choice:
3
balance: 52500

menu
1. deposit 2. withdraw 3. compute Interest
4. display
bas
enter the amount in
20000

menu
1. display 2. withdraw 3. compute
Interest 4. display

enter the choice:
4

name: Sanvi acno: 300 type:
savings balance: 76125.0

enter the name
Sanvi
enter: account number
300
enter initial balance:
3000

Quick Work
Page No.:
Date:
M T W T F S S

enter the type (current / savings):
current

menu
1: deposit 2: withdraw 3: display
enter the choice:
1
enter the amount:
10000

Savi
1BM22CS245

enter the choice:
3

name: savi acc no: 300 type: current
balance: 40000

menu
1: deposit 2: withdraw 3: display
enter the choice
2
enter the amount:
20000

menu
1: display 2: withdraw 3: display
enter the choice:
3

name: savi acc no: 300 type: current
balance: 20000

16/1/2024

Program 6:

Strings:

1. Demonstrate various string constructor with proper java prog.

o/p:

Dimensions are 10.0 by 14.0 by 12.0
Box b = Dimensions are 10.0 by 14.0 by 12.0

2. Demonstrate string length, string literal, string concat.

~~Date Prog~~

o/p:

Hello world // string ~~length~~ literal

// String length
Length of the string: Hello world = 13

// string concatenation

Concatenated string: Hello world

3. Demonstrate tostring()

o/p:

dimensions are 10.0 by 14.0 by 12.0
box b: Dimensions are 10.0 by 14.0 by 12.0.

Quick Work
Page No.:
Date:
M T W T F S S

4. using getchars(), extract Bmsce from "welcome to Bmsce college"
o/p:
me to b

5. Demonstrate getbytes(), toCharArray()
o/p:

//getbytes
o/p:

72
101
65
108
111
32
66
109
115
99
101

//toCharArray
welcome to bmsce college

6. Write the following output & write the java prog. using String functions
Bmsce equals Bmsce → true
Bmsce equals college → false
Bmsce equals BMSCE → false
Bmsce equalsIgnoreCase BMSCE → true

o/p:
//input = Bmsce
true

Sawi Nadiga
1BM22CS245

7. Using region matches() find the substring "Bmsce college" from the string "welcome to Bmsce college of Engineering", if matches display substring is matched otherwise display not matched.

O/p:
I/input → Bmsce college.
O/p:
substring is not matched.

8. Demonstrate startwith() to give o/p true and false.

O/p:
it is true.

9. Demonstrate endswith() to give o/p true & false.

O/p:
it is false.

10. Demonstrate a java prog to show o/p. for equals() vs. ==

O/p:
Hello equals Hello → true
Hello == Hello → false

Quick Work
Page No.:
Date:
M T W T F S S

11. Write a java prog. to perform sorting for alphabets using compareto().

o/p:

apple
bake
cat
dog
ent
frd
gun
hen
ice
jug
kite
list
man
net
orange
parrot
queveen
ring
stem
tree
umbrella
van
watch
xmas
yatch
zee

Sanvi Nadiga
1BM22CS264

12. Write a java prog. to perform addition of two no. using compareTo()

O/P:
1
2
3
4
5
6
7
8
9
10

13. Write a java prog. using substring, indexOf()

O/p:
That was a test
that was too.

14. Hello world.

15. O/p: Welcome

16. O/p: Hello friends

17. Registration name: 26
Full name: Alice bob
Semester: 2
CGPA: 8.5

Quick Work

Page No.:
Date:
M T W T F S S

(d) i) 7.6
8.5
9.2

ii) Alice
Bob
John

18. o/p: 11String ~~toUpper~~ length
o/p: Hello

11 string ~~length~~ charAt
o/p: H

11 string setCharAt:
o/p: HXllo

11 getChars:
o/p: World.

11 append:
o/p: Hello, World!

11 insert:
o/p: Hello world.

11 reverse:
o/p: olleh

11 delete:
o/p: Hello!

// deleteCharAt :
o/p: Hllo

// replace :
o/p: Hello, universe

// substring :
o/p: World.

19. o/p: Eagle details :
Eagle soars high in the sky
Eagle screeches loudly.

Hawk details :
Hawk glides through the sky
Hawk emits sharp cry.

20. o/p: Circle details :
Area = 3.14
Perimeter = 31.42

Triangle details
Area = 6
Perimeter = 12

Lab 7

Quick Work
Page No.:
Date:
M T W T F S S

Q: Design a package called maths having the class called number (add & sub methods) Implement a simple class called maths demo to use maths (outside package maths) that makes use of package provided by maths.

Create a package CIE which has 2 classes "student" and internal. The class student has members like usn, name, sem. The class "internals" derived from student has an array that stores the internal marks scored in 5 courses of the current sem of the student. This class has an array that stores the SEE marks scored in 5 courses of the current sem of the student. Import the 2 packages in a file that declares the final marks of n students in all 5 courses.

```
package CIE;
import java.util.*;
public class student{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails()
    {
```

```
Scanner sc = new Scanner (System.in);
System.out.println ("Read usn");
usn = sc.next();
System.out.println ("Read name");
name = sc.next();
System.out.println ("Read sem");
sem = sc.nextInt();
}
public void displayStudentDetails()
{
    System.out.println ("usn: " + usn);
    System.out.println ("name: " + name);
    System.out.println ("sem: " + sem);
}
}
```

```
package cie;
import java.util.*;
public class Internals extends Student{
    protected int marks[] = new int[5];

    public void inputCIEmarks()
    {
        Scanner sc = new Scanner (System.in);
        for (int i = 0; i < 5; i++)
        {
            System.out.println ("Enter Marks");
            marks[i] = sc.nextInt();
        }
    }
}
```

```
package SEE;
import CIE.internals;
import java.util.Scanner;
public class Externals extends Internals
{
    protected int marks[];
    protected int finalmarks[];
    public Externals()
    {
        marks = new int[5];
        finalmarks = new int[5];
    }
    public void inputSEEMarks()
    {
        Scanner sc = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.print("Subject " +
                (i+1) + " marks: ");
            marks[i] = sc.nextInt();
        }
    }
    public void displayFinalMarks()
    {
        displayStudentDetails();
        for(int i=0; i<5; i++)
        {
            System.out.println("Subject " +
                (i+1) + ": " + finalmarks
                [i]);
        }
    }
}
```

```
import SEE.Externals;

class main{
public static void main(String args[])
{
    int numOfStudents = 2;
    Externals finalMarks[] = new
    Externals[numOfStudents];

    for(int i=0; i<numOfStudents; i++)
    {
        finalMarks[i] = new Externals();
        finalMarks[i].inputStudentDetails();

        System.out.println("Enter CIE marks");
        finalMarks[i].inputCIEMarks();

        System.out.println("Enter SEE marks");
        finalMarks[i].inputSEEmarks();
    }

    System.out.println("Display data:");
    for(i:= 0; i < numOfStudents; i++)
    {
        finalMarks[i].calculateFinal();
        finalMarks[i].displayFinalMarks();
    }
}
}
```

Quick Work

Page No.:
Date:
S M T W T F S S

O/p:

read usn of student:
1BM22CS245
read name of student:
Sanvi
read sem of student:
3
Enter 5 marks:
40
45
50
42
43

Sanvi Nadiga
1BM22CS245

Enter CIE marks:
20
50
30
40
45

Student details:
final marks: math: 60
Eng: 95

24/1/2024

LAB 8

import java.util.*;

Write a prog. that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception wrongAge() when input age < 0. In Son class, implement a constructor that takes both Father & Son's age & throws an exception if Son's age is >= father's age.

```
import java.util.*;

Class WrongAge extends Exception{
    public WrongAge(String s)
    {
        super(s);
    }
}
Class Father{
    private int age;
    public Father(int age) throws
                    WrongAge {
```

Quick Work
Page No.:
Date:
M T W T F S S

if (age < 0) {
throw new WrongAge
("Age cannot be negative");
}
this.age = age;
}
class Son extends Father {
private int sonAge;
public Son(int fatherAge, int
sonAge) throws
WrongAge
{
super(fatherage);
if (sonAge >= fatherAg
{
throw new WrongAge
("Son's age
cannot be greater th
or equal to father's
age");
}
this.sonAge = sonAge;
}
}
public class Father main
{

```
public static void main(String[] args){
    try{
        Son son = new Son(-19, 85);
    }catch(WrongAge e){
        System.out.println(e.getMessage());
    }
}
```

O/p:
when input values are -19 & 85
O/p:
Age cannot be negative

when input values are 35 & 55
O/p:
Son's age cannot be greater than or equal to father's age

when input values are 35 & 35
O/p:
Son's age cannot be greater than or equal to father's age

Quick Work
Page No.:
Date:
M T W T F S S

when input values are 55 & 35

o/p:

Age is valid

30/4/2024

Sanvi Nadiga
1BM22CS245

Lab Prog 8:

Write a program which creates 2 threads one thread displaying "BMS college of Engineering" once every 10 seconds another displaying "CSE" once every 2 seconds

```
class bms extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++)
        {
            System.out.println("bms college of engineering " + i);
            try
            {
                Thread.sleep(10000);
            }
            catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}

class cse extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++)
        {
            System.out.println("cse " + i);
```

```
try
{
    Thread.sleep(9000);
}
catch (InterruptedException e)
{
    e.printStackTrace();
}
}
}

class ThreadMain{
    public static void main(String args[])
    {
        bms obj1 = new bms();
        cse obj2 = new cse();
        obj1.start();
        obj2.start();
    }
}
```

O/P:

bms college of engineering 1
cse 1
cse 2
cse 3
cse 4
cse 5
bms college of engineering 2
cse 6

Quick Work
Page No.:
Date:
M T W T F S S

cse 7
cse 8
cse 9
cse 10
bmsce college of engineering 3
bme college of engineering 4
bms college of engineering 5

Sami Nadiga
1BM22CS245

Quick Work
Page No.:
Date:
M T W T F S S

Lab prog 10

Demonstrate Inter Process Communication & deadlock.

```
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
        while (!valueSet)
        {
            try
            {
                System.out.println("Consumer
                    waiting");
                wait();
            }
            catch (InterruptedException e)
            {
                System.out.println("Exception
                    caught");
            }
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
    }
}
```

```
synchronized void put(int n)
{
while(valueSet)
{
try
{
System.out.println("Producer waiting");
wait();
}
catch(InterruptedException e)
{
System.out.println("Exception caught");
}
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("Intimate consumer\n");
notify();
}

class Producer implements Runnable
{
Q q;
Producer(Q q)
{
```

```
        this.q = q;
        new Thread (this, "Producer").start();
    }
    public void run() {
        int i = 0;
        while (i < 5) {
            q.put(i++);
        }
    }
}

class Consumer implements Runnable {
    Q q;
    Consumer (Q q) {
        this.q = q;
        new Thread (this, "Consumer").start();
    }
    public void run() {
        int i = 0;
        while (i < 5) {
            int r = q.get();
            System.out.println("Consumed : " + r);
            i++;
```

```
    }
  }
}
class PCFixed
{
  public static void main (String
                           args[ ])
  {
    Q q = new Q( );

    new Producer (q);
    new Consumer (q);
    System.out.println(" Press
      Control - C to stop. ");

  }

}
```

O/P:

```
put : 0
Intimate consumer

producer waiting
Press control - C to stop.
Got : 0
Intimate Producer.

Put : 1
Intimate Consumer
```

o/p

Producer waiting
consumer: 0
Put: 1
Intimate Producer

consumed: 1
put: 2
Intimate Producer

Producer waiting
consumed: 0
Put: 1
Intimate Producer

consumer: 1
put: 2
Intimate consumer

Producer waiting
Put: 2
Intimate Producer

consumed: 2
put: 3
Intimate consumer

consumer: 3
put: 4
Intimate consumer

Name: Nadiga
[illegible]

Got : 4
Intimate Producer

Consumed : 4

6/2/2024

18/7/24

Quick Work
Page No.:
Date:
M T W T F S S

Lab 10 (B)

Demonstrate Deadlock.

```
class A {
    synchronized void foo(B b)
    {
        String name = Thread.currentThread
                      ().getName();
        System.out.println(name + " entered
                           A.foo");
        try {
            Thread.sleep(1000);
        }
        catch (Exception e)
        {
            System.out.println("A Interrup
                               ted");
        }
        System.out.println(name + " trying
                           to call B.last()");
        b.last();
        System.out.println(name + "finished
                           executing A.foo");
    }
    synchronized void last()
```

```
        System.out.println(" Inside A.last");
    }
}

class B {

    synchronized void bar(A a)
    {
        String name = Thread.currentThread().
                                getName();
        System.out.println(name + " entered
                                B.bar");

        try
        {
            Thread.sleep(1000);
        }
        catch (Exception e)
        {
            System.out.println("B interrupted");
        }
        System.out.println(name + " trying
                                to call A.last()");

        a.last();

    System.out.println(name + " finished
                executing B.bar");
    }
```

Quick Work
Page No:
Date:
M T W T F S S

```
synchronized void last()
{
    System.out.println("Inside B.last");
}

class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();

    Deadlock()
    {
        Thread.currentThread().setName
            ("MainThread");

        Thread t = new Thread(this,
            "RacingThread");

        t.start();
        a.foo(b);
        System.out.println("Back in
                mainthread");
    }

    public void run()
    {
        b.bar(a);
        System.out.println("Back in other
                thread");
    }
```

Quick Work
Page No.:
Date:
M T W T F S S

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

O/p:

```
MainThread entered A.foo
RacingThread entered B.bar
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Back in mainThread
RacingThread trying to call A.last()
Inside A.last
Back in Other thread.
```

13/2/2024

Sanvi Nadiga
1BM22CS245

Quick Work
Page No.:
Date:
M T W T F S S

9/12/24 Lab - 10

10. Write a program that creates a user interface to perform integer division. The user enters 2 numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the result field when the divide button is clicked. If Num1 and Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an ArithmaticException display the exception in a message dialog box.

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

public class SwingDemo {
    public SwingDemo() {

        JFrame jfrm = new JFrame("Divider app");
        jfrm.setSize(275, 150);
        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(blab);
jfrm.add(ansLab);

button.addActionListener(new
                ActionListener()
{
  public void actionPerformed
        (ActionEvent evt)
  {
    try{
      int a = Integer.parseInt
```

```
(ajft.getText());
int b = Integer.parseInt(bjft.
                    getText());
if (b==0){
        throw new ArithmeticException
            ("Division by zero
             not allowed");
}
int ans = a/b;
alab.setText("A = " + a);
blab.setText("B = " + b);
anslab.setText("Ans = " + ans);
}
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!");
}
}
});
jfrm.setVisible(true);
}
public static void main(String args[])
{
SwingUtilities.invokeLater(new
Runnable()
{
public void run()
{
new SwingDemo();
}
});
}

Quick Work
Page No.:
Date:
M T W T F S S

O/p:

Samiri Nadiga
1BM22CS245

Divider app

Enter the divider and divident:

22 | 2

calculate A = 22 B = 2 Ans: 11

Definitions:

JFrame:
A class in Java that has its own modules and constructors.

FlowLayout():
creates a flow layout with centered alignment and a default 5 unit horizontal and vertical gap.

jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE):
to terminate on close.

Jbutton("Calculate:"):
Button with text calculate inside.

JLabel():
To give labels to the objects.

jtm.add(err):
To add error object that has the label.

ActionListener():
Defines what should be done when a user performs a certain operation

getText():
This method returns the string as a result.

setText():
This method recieves a String.

SetVisible():
if you set it to true, it means you want that thing to be visible in your screen.

JTextField:
A lightweight component that allows the editing of a single line of text.

20/2/2024