

JAVA END OF SEMESTER LAB REPORT

Name: Rushi Hundiwala
USN: 1BM22CS24

LAB→ HelloWorld.java

```
class HelloWorld {  
    public static void main (String [] args) {  
        System.out.println ("Hello, World!");  
    }  
}
```

→ Quadratic.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 ("enter the co-eff  
        of . a,b,c");  
        a = s.nextInt();  
        b = s.nextInt();  
        c = s.nextInt();  
    }  
}
```

```
void compute()
```

{

```
    while(a==0)
```

{

```
        System.out.println("Not a quad.eq");
```

```
        System.out.println("Enter a non-zero
```

```
value for a:");
```

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

```
a=s.nextInt();
```

}

$$d = b^2 - 4ac;$$

```
if(d==0)
```

$$r_1 = -b / (2a);$$

```
System.out.println("roots are real & equal");
```

```
System.out.println("root1=root2=" + r1);
```

}

```
else if(d>0){
```

~~$$r_1 = ((-b) + \sqrt{d}) / (2a);$$~~

~~$$r_2 = ((-b) - \sqrt{d}) / (2a);$$~~

~~```
System.out.println("real & 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("root2 = " + r1 + " - i" + r2);
}

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

Output

→ Enter the coeff of a,b,c :

1 -3 2

Roots are real & distinct
Root 1 = 2.0, Root 2 = 1.0

LAB 2

```
import java.util.Scanner;
class Subject {
    int subjectMarks; // for
    int credits; // entering marks
    int grade; // (1+1) + Total
    Subject() { // from width
        // Total / credits. [i] to due
        // time - SGP A
    }
}
class Student {
    String name; // for (i) to id
    String USN; // for (i)
    double SGPA; // for (i) to id
    Scanner s; // for (i)
    Subject[] subject; // for (i)
    Student() {
        int i;
        subject = new Subject[8];
        for (i = 0; i < 8; i++)
            subject[i] = new Subject();
        s = new Scanner(System.in);
    }
}
void getStudentDetails() {
    System.out.println("Enter Name");
    name = s.nextLine();
    System.out.println("Enter USN:");
    USN = s.nextLine();
}
```

```
mo: '
int
n i
n j
J
void getMarks() {
    int i;
    for (i=0; i<8; i++) {
        System.out.println("Enter marks for
            subject " + (i+1) + ": ");
        subject[i].subjectMarks = s.nextInt();
        subject[i].credits = 4;
        if (subject[i].subjectMarks >= 90)
            subject[i].grade = 10;
        else if (subject[i].subjectMarks >= 80)
            subject[i].grade = 9;
        else if (subject[i].subjectMarks >= 70)
            subject[i].grade = 8;
        else if (subject[i].subjectMarks >= 60)
            subject[i].grade = 7;
        else if (subject[i].subjectMarks >= 50)
            subject[i].grade = 6;
        else
            subject[i].grade = 0;
    }
}
```

```
void computeSPPA() {
```

```
double totalCredits = 0; : twqtwD
double weightedTotal = 0;
for (int i = 0; i < 8; i++) { return
    totalCredits += subject[i].credits;
    weightedTotal += (subject[i].grade * subject[i].credits);
}
SGPA = weightedTotal / totalCredits;
System.out.println("SGPA : " + SGPA);
}

public class main {
    public static void main (String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}
```

}

- Sx20
Sx25
19.12.20

Output

Output : $i = 2 + 3 \times 0 \rightarrow i = 5$ output 5
 $i = 1 + 0 \times 3 \rightarrow i = 1$ output 1

Enter Name : *(for example: 8 > i ; o = i + n) rot*

John

Enter USNI:

12345

Enter Marks for subject :-

25

85 „(z+Nbr) Int + „Int subj 2: = Aq, 2

78 + " Almond) mit einer kleinen Menge

Enter marks for subject 3:

92

Enter marks for subject 4: Silding

67 ~ 10' m ~ 10' v ~ 5' s

Enter marks for subject

55

55
Entry () marbles 10^v subject 6:

73

三

11.00 - 7.625

SGPA : - 7.0 - 10 FORM 12

LAB 3

IBM22 CS 224

LAB - 3Rushi Hundiwala

DATE:

PAGE:

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

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

```
    Books (String name, String author,  
           int price, int numPages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;
```

```
    public String toString () {  
        String bookDetails;  
        bookDetails = "Book name : "  
            + this.name + "\n" + "Author: " +  
            this.author + "\n" + "Price: " + this.price  
            + "\n" + "Number of Pages : " + this.  
            numPages + "\n";  
        return bookDetails;  
    }
```

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

```
        int n = s.nextInt();
```

```
        Books b [] = new Books [n];
```

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

~~System.out.println ("Enter details for
Book " + (i + 1));~~

~~System.out.println ("Enter book name: ");~~

~~String name = s.next();~~

~~System.out.print ("Enter author name: ");~~

~~String author = s.next();~~

~~System.out.println ("Enter price: ");~~

~~int price = s.nextInt();~~

~~System.out.print ("Enter number of pages: ");~~

~~int numPages = s.nextInt();~~

~~b [i] = new Books (name, author
, price, num Pages);~~

}

~~System.out.println ("Book Details: ");~~

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

```
    System.out.println ("Book" +(i+1)  
    +"Details:" +"\n"+ b[i]);  
}
```

Output :-

2

| | | | |
|--------|----------|----|-----|
| Book 1 | Author 1 | 20 | 300 |
| Book 2 | Author 2 | 25 | 400 |

Enter book name: Book1

Enter author name: Author1

enter price: 20

enter number of pages: 300

Enter details for Book 2

Enter book name: Book2

enter author name: Author2

enter price: 25

enter number of price: 400

Book Details:

Book 1 Details:

Book name: Book1

Author name: Author1

Price: 20

Number of Pages: 300

DATE:

PAGE:

~~Book 2 Details~~

~~Book name: Book 2~~

~~Author Name: Author 2~~

~~Price: 25~~

~~Number of Pages: 400~~

~~8/20/2023~~

LAB 4

WEEK - 4

DATE: PAGE:

```
import java.util.Scanner;  
  
public class InputScanner {  
    private static final Scanner scanner = new Scanner(System.in);  
  
    public static int  
    getIntInput (String prompt) {  
        System.out.print(prompt);  
        return scanner.nextInt();  
    }  
}
```

```
public class Rectangle extends Shape {  
    @Override  
    public void printArea () {  
        System.out.println ("Area of  
        Rectangle : " + (side1 * side2));  
    }  
}
```

```
public class Triangle extends Shape {  
    @Override  
    public void printArea () {  
        System.out.println ("Area of  
        Triangle : " + (0.5 * side1 * side2));  
    }  
}
```

public class Circle extends Shape;
 @ override
 public void printArea() {
 System.out.println("Area of
 Circle: " + (Math.PI * side2 * side1));
 }
 public class MainClass {
 public static void main(String[] args) {

Rectangle rectangle = new
 Rectangle();
 rectangle.side1 = getIntInput()
 enter length of rectangle :);
 rectangle.side2 = getIntInput()
 "enter width of rectangle : ");
 rectangle.printArea();

Triangle triangle = new Triangle()
 triangle.side1 = getIntInput()
 enter base of triangle :);
 triangle.side2 = getIntInput()
 "enter height of triangle : ");
 triangle.printArea();

Circle circle = new Circle();
circle.side1 =
getInput("Enter radius of
circle: ");

circle.printArea();

Output

Enter length of rectangle: 5
Enter width of rectangle: 3
Area of Rectangle: 15

Enter base of triangle: 4
Enter height of triangle: 6
Area of Triangle: 12.0

Enter radius of circle: 2
Area of circle: 12.566370614359172

B
02/01/24

LAB 5

LAB-5

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

```
class Account {
```

```
    String customerName;
```

```
    int accountNumber;
```

```
    String accountType;
```

```
    double balance;
```

```
    Account (String name, int accNumber  
            , String accType) {
```

```
        customerName = name;
```

```
        accountNumber = accNumber;
```

```
        accountType = accType;
```

```
        balance = 0.0;
```

```
}
```

```
    void deposit (double amount) {
```

```
        balance = balance + amount;
```

```
}
```

```
    void displayBalance () {
```

~~System.out.println("Balance for
account " + accountNum + " is " + balance);~~

```
}
```

```
}
```

```
class CurAcct extends Account {  
    double minBalance; // trophi  
    double serviceCharge; // trophi  
    CurAcct (String name, int accNumber) {  
        super (name, accNumber, "Current");  
        minBalance = 1000.0; // point 2  
        serviceCharge = 10.0; // point 2  
    }  
}
```

```
void checkMinBalance () {  
    if (balance < minBalance) {  
        balance += balance - serviceCharge;  
        System.out.println ("Service charge  
imposed. New balance: $" +  
        balance);  
    }  
}
```

```
class SavAcct extends Account {
```

```
    double interestRate;
```

Sav Acct (String name, int accNumber)

{

super (name, accNumber, "Savings");
interestRate = 0.05;

}

void computeInterest () {

double interest = balance * interestRate;

balance = balance + interest;

System.out.println ("withdraw" + balance);

}

else {

System.out.println ("insufficient funds for
withdrawal");

{

}

public class Bank {

public static void main (String [] args) {

Scanner s = new Scanner (System.in);

currAcct currAcct = new currAcct ("
John Doe", 123456);

currAcct . deposit (1500.0);

currAcct . display Balance();

currAcct . checkMinBalance();

SavAcct savAcct = new SavAcct ("
Jane Smith", 789012);

savingsAccount . deposit (2000.0);

savingsAccount . display Balance();

savingsAccount . compute Interest();

Output

Balance for account 123456:

1500.0

Balance for account 123456: 1490.0

Balance for Account 789012: 2000.0

Interest computed . New Balance : \$2100.0

LAB 6

23/08/24

WEEK-6

DATE:

PAGE:

PAGE:

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 s = new Scanner(System.in);

System.out.println("give usn");

usn = s.nextLine();

System.out.println("give name");

name = s.nextLine();

System.out.println("give sem");

sem = s.nextInt();

public void displayStudentDetails() {

System.out.println("The usn is : " + usn);

System.out.println("The name is : " + name);

System.out.println("The sem is : " + sem);

```
package CIE;  
import java.util.Scanner;  
  
protected int marks[] = new int[5];  
  
public void input(CIE amarks){  
    Scanner s = new Scanner(System.in);  
  
    for (int i=0; i<5; i++) {  
        System.out.println("Enter marks for course " + (i+1) + ":");  
        marks[i] = s.nextInt();  
    }  
}
```

```
package SEE;
```

```
import CIE.Internals;  
import java.util.Scanner;  
  
public class External extends Internals {  
  
    protected int marks[];  
    protected int finalMarks[];  
}
```

public External () {

marks = new int[5];

finalMarks = new int[5];

}

public void input SEEmarks() {

Scanner s = new Scanner (System.in);

for (int i=0; i<5; i++) {

System.out.println ("Subject" + (i+1) + "marks :");

marks[i] = s.nextInt();

marks[i] = s.nextInt();

public void calcFinalMarks() {

for (i=0; i<5; i++) {

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

}

package SEFE;
public class Externals extends Internals

public Externals;

public class Main {

public static void main(String args[])

{

for (i = 0; i < num of students; i++) {

finalMarks[i] = new Externals();

finalMarks[i] = new Externals();

finalMarks[i].input(Externals[i]);

System.out.println ("display data!");

for (int i = 0; i < num of students; i++) {

finalMarks[i] = calcFinalMarks();

3

8/3/14
V

Output → hab. 16

student 1 marks 100

Enter marks of

subject 1 : 30

2 : 50

3 : 40

4 : 20

5 : 10

Language

Student 2

Enter marks of

subject 1 : 30

2 : 70

3 : 60

4 : 80

5 : 90

CIE:

sub 1 marks : 30

sub 2 marks : 50

sub 3 marks : 40

sub 4 marks : 20

sub 5 marks : 10

SEE

| | | | | |
|-----|---|-------|---|----|
| sub | 1 | marks | : | 30 |
| " | 2 | " | : | 20 |
| " | 3 | " | : | 60 |
| : | 4 | : | : | 80 |
| : | 5 | : | : | 90 |

LAB 7

Lab - 7

DATE:

PAGE:

```
PAGE: import java.util.Scanner;  
class WrongAge extends Exception {  
    public WrongAge() {  
        super("message: \"Age can't be neg\"");  
    }  
    public WrongAge(String message) {  
        super(message);  
    }  
}  
class InputScanner {  
    public static int readInt() {  
        try (Scanner scanner = new Scanner(  
            System.in)) {  
            return scanner.nextInt();  
        }  
    }  
}  
class Father extends InputScanner {  
    protected int fatherAge;  
    public Father() throws WrongAge {  
        System.out.print("enter father's age : ");  
        fatherAge = readInt();  
        if (fatherAge < 0) {  
            throw new WrongAge("Age cannot be  
negative");  
        }  
    }  
}
```

PAGE

```
public void display () {
    cout << "father's age : " -> fatherAge;
}

class Son extends Father {
protected int sonAge;

public Son() throws WrongAge {
    super();
    cout << "enter sons age : ";
    sonAge = readInt();
    if (sonAge >= super.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");
    }
}

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

public class Exceptions {

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

```
try {
```

```
Son son = new Son();
```

```
son.display(); }
```

```
catch (WrongAge e) {
```

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

3
3

~~Java~~

~~Java~~
~~30.01.09~~

Output

age

```
Enter father's age : 45
```

```
Enter son's age : 50
```

Error: Son's age cannot be
greater than father's age.

LAB 8

WEEK - 8

DATE: PAGE:

PAGE:

```
class DisplayMessage extends Thread {
```

```
    private String message;  
    private int interval;
```

```
    public DisplayMessage (String message,  
                          int interval) {
```

```
        this.message = message;  
        this.interval = interval;
```

@ Override

```
    public void run () {
```

```
        while (true) {
```

```
            System.out.println (message);
```

```
        try {
```

```
            Thread.sleep (interval * 1000);
```

```
        catch (InterruptedException e) {
```

```
            e.printStackTrace ();
```

```
}
```

```
    }
```

```
    public class Threading {
```

public static void main (String [] args)

DisplayMessage thread1 = new DisplayMessage
("BMS College of Engineering", 10);

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

thread1.start();

thread2.start();

Output - Lab [8]

CSE

BMS College of Engineering

CSE

BMS College of Engineering

CSE

BMS College of Engineering

CSE

LAB 9

lab 9

DATE:

PAGE:

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

```
class Swing Demo {
```

```
Swing Demo() {
```

```
// create jfrm container
```

```
JFrame jfrm = new JFrame("Dividers 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 divisor & dividend.");
```

```
// add text field for both  
numbers
```

```
JTextField
```

```
JTextField
```

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

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

```
// calc button
```

JButton button = new JButton("Calculate")

11 labels

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

11 add in order :)

App:
jfrm.add(err);
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

Action Listener] = new ActionListener()

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

```
ajtf.addActionListener(1);
bjtf.addActionListener(1);

button.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        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("A = " + a);
            blab.setText("B = " + b);
            anslab.setText("Ans = " + ans);
            err.setText("");
        } catch (ArithmeticException e1) {
            err.setText("Division by zero");
        }
    }
});

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

1) Create frame on event
dispatching thread

SwingUtilities.invokeLater(new Runnable)

```
public void run() {
    new SwingDemo();
}
```

Output

| | |
|--|----------------------|
| Enter the divisor and dividend | |
| <input type="text"/> | <input type="text"/> |
| <input type="button" value="Calculate"/> | |

$$a = 10$$

$$b = 2$$

$$\underline{\text{ans}} = 5$$

~~Java GUI~~

LAB 10

WEEK-10

DATE:

PAGE:

public class DeadlockExample {

 public static void main (String[] args) {

 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 Object lock1 = new Object();
private Object lock2 = new Object();

public void method1() throws
InterruptedException {

synchronized (lock2) {

System.out.println("Method 1 acquired
lock2");
Thread.sleep(1000);

synchronized (lock1) {

System.out.println("Method 1 acquired
lock1");

OP
X3
B6.0

Output - Lab 110

Method 1 . acquired lock1
Method 2 acquired Lock2

✓
SFT
13.02.2012

SOURCE FILES FOR JAVA LAB PROGRAMS

LAB 1

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

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

```
        System.out.println("Root1 = " + r1 + " + i" + r2);
        System.out.println("Root2 = " + r1 + " - i" + r2);
    }
}
}

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

LAB 2

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    // Constructor
    Subject() {
        // Default constructor
    }
}

class Student {
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject[] subject;

    // Constructor
    Student() {
        int i;
        subject = new Subject[8]; // Corrected the array size to match the number of subjects
        for (i = 0; i < 8; i++)
            subject[i] = new Subject();
        s = new Scanner(System.in);
    }

    // Method to get student details
    void getStudentDetails() {
        System.out.println("Enter Name: ");
        name = s.next();
        System.out.println("Enter USN: ");
        usn = s.next();
    }

    // Method to get marks and calculate grade
```

```

void getMarks() {
    int i;
    for (i = 0; i < 8; i++) {
        System.out.println("Enter marks for Subject " + (i + 1) + ": ");
        subject[i].subjectMarks = s.nextInt();

        // Assuming credits are fixed at 4 for each subject
        subject[i].credits = 4;

        // Calculate grade based on marks
        if (subject[i].subjectMarks >= 90) subject[i].grade = 10;
        else if (subject[i].subjectMarks >= 80) subject[i].grade = 9;
        else if (subject[i].subjectMarks >= 70) subject[i].grade = 8;
        else if (subject[i].subjectMarks >= 60) subject[i].grade = 7;
        else if (subject[i].subjectMarks >= 50) subject[i].grade = 6;
        else subject[i].grade = 0; // Assuming 0 grade points for marks below 50
    }
}

// Method to compute SGPA
void computeSGPA() {
    double totalCredits = 0;
    double weightedTotal = 0;

    for (int i = 0; i < 8; i++) {
        totalCredits += subject[i].credits;
        weightedTotal += (subject[i].grade * subject[i].credits);
    }

    SGPA = weightedTotal / totalCredits;

    System.out.println("SGPA: " + SGPA);
}
}

public class Main {
    public static void main(String[] args) {
        // Declare and initialize Student object
        Student s1 = new Student();

        // Call methods to get details, marks, and compute SGPA
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}

```

}

LAB 3

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

import java.io.*;

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 - Name: " + 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("Rushi, 1bm22cs224");

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

LAB 4

```
/* Rushi Hundiwala */
/* 1BM22CS224 */
```

```
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

NAME: Rushi Hundiwala
USN:1BM22CS224

```
import java.util.Scanner;

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

    Account(String name, int accNumber, String accType) {
        customerName = name;
        accountNumber = accNumber;
        accountType = accType;
        balance = 0.0;
    }

    void deposit(double amount) {
        balance += amount;
    }

    void displayBalance() {
        System.out.println("Balance for account " + accountNumber + ": $" + balance);
    }
}

class CurAcct extends Account {
    double minBalance;
    double serviceCharge;

    CurAcct(String name, int accNumber) {
        super(name, accNumber, "Current");
        minBalance = 1000.0; // Example minimum balance for current account
        serviceCharge = 10.0; // Example service charge for falling below minimum balance
    }

    void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
        }
    }
}
```

```

        System.out.println("Service charge imposed. New balance: $" + balance);
    }
}
}

class SavAcct extends Account {
    double interestRate;

    SavAcct(String name, int accNumber) {
        super(name, accNumber, "Savings");
        interestRate = 0.05; // Example interest rate for savings account
    }

    void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest computed. New balance: $" + balance);
    }

    void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal successful. New balance: $" + balance);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
    }
}

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

        // Example usage
        CurAcct currentAccount = new CurAcct("John Doe", 123456);
        currentAccount.deposit(1500.0);
        currentAccount.displayBalance();
        currentAccount.checkMinBalance();

        SavAcct savingsAccount = new SavAcct("Jane Smith", 789012);
        savingsAccount.deposit(2000.0);
        savingsAccount.displayBalance();
        savingsAccount.computeInterest();
    }
}
```

```
// You can add more functionality or create additional accounts as needed
}
}
```

LAB 6

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

package CIE;

import java.util.Scanner;

public class Student {

    protected String usn;
    protected String name;
    protected int sem;

    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN:");
        usn = s.nextLine();
        System.out.println("Enter name:");
        name = s.nextLine();
        System.out.println("Enter semester:");
        sem = s.nextInt();
    }

    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

package CIE;

import java.util.Scanner;

public class Internals extends Student {

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

    public void inputCIEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter CIE marks for 5 courses:");
        for (int i = 0; i < 5; i++) {
```

```

        System.out.print("Course " + (i + 1) + ": ");
        marks[i] = s.nextInt();
    }
}
}

package SEE;

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

public class Externals extends Internals {

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

    public void inputSEEMarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter SEE marks for 5 courses.");
        for (int i = 0; i < 5; i++) {
            System.out.print("Course " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }

    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] / 2 + super.marks[i];
        }
    }

    public void displayFinalMarks() {
        displayStudentDetails();
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}

// Main.java
import SEE.Externals;

public 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("Displaying data:\n");

for (int i = 0; i < numOfStudents; i++) {
    finalMarks[i].calculateFinalMarks();
    finalMarks[i].displayFinalMarks();
}
}
```

LAB 7

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

import java.util.Scanner;

class WrongAge extends Exception {
    public WrongAge() {
        super("Age cannot be negative");
    }

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

class InputScanner {
    public static int readInt() {
        Scanner scanner = new Scanner(System.in);
        return scanner.nextInt();
    }
}

class Father extends InputScanner {
    private int fatherAge;

    public Father() throws WrongAge {
        System.out.print("Enter father's age: ");
        fatherAge = readInt();
        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

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

class Son extends Father {
    private int sonAge;

    public Son() throws WrongAge {
```

```
super(); // Call the constructor of the base class (Father)
System.out.print("Enter son's age: ");
sonAge = readInt();
if (sonAge >= super.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");
}
}

public void display() {
    super.display(); // Call the display method of the base class (Father)
    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("Error: " + e.getMessage());
        }
    }
}
```

LAB 8

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

class DisplayMessage extends Thread{
private String message;
private int interval;

public DisplayMessage(String message,int interval){
this.message=message;
this.interval=interval;
}

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

public class Threading{
public static void main(String[] args){
    DisplayMessage thread1 = new DisplayMessage("BMS College of Engineering", 10);
    DisplayMessage thread2 = new DisplayMessage("CSE", 2);
    thread1.start();
    thread2.start();
}
}
```

LAB 9

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

import java.awt.*;
import java.awt.event.*;
import javax.swing.JPanel;
import java.util.Random;

public class ButtonDrag extends Frame implements ActionListener {
    int n = 3;
    int m = n * n;
    boolean clicked = false, doneFlag = false;
    String cLabel;
    int cl;
    JPanel buttonPanel = new JPanel();
    JPanel optionPanel = new JPanel();
    Button[] b = new Button[n * n];
    Button start, reset, restart;
    String msg = "";
    TimeCalc total;
    int totalTime;

    public ButtonDrag() {
        addWindowListener(new MyWindowAdapter());
        setLayout(new BorderLayout());
        buttonPanel.setLayout(new GridLayout(n, n));
       setFont(new Font("Arial", Font.BOLD, 24));
        buttonPanel.setSize(300, 300);
        buttonPanel.setEnabled(false);
        optionPanel.setLayout(new FlowLayout());
        add(buttonPanel, BorderLayout.CENTER);
        add(optionPanel, BorderLayout.SOUTH);
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                int k = i * n + j;
                if (k > 0) {
                    buttonPanel.add(b[k] = new Button("'" + k));
                }
            }
        }
        buttonPanel.add(b[0] = new Button("9"));
        for (int i = 0; i < m; i++) {
```

```

        b[i].addActionListener(this);
    }
    optionPanel.add(reset = new Button("Reset"));
    optionPanel.add(start = new Button("Start"));
    optionPanel.add(restart = new Button("Restart"));
    start.addActionListener(this);
    reset.addActionListener(this);
    restart.addActionListener(this);
    restart.setEnabled(false);
    reset.setEnabled(false);
    Component[] com = buttonPanel.getComponents();
    for (int a = 0; a < com.length; a++)
        com[a].setEnabled(false);
}
}

public void actionPerformed(ActionEvent ae) {

    if (ae.getSource() == start && (!doneFlag)) {
        Component[] com = buttonPanel.getComponents();
        for (int a = 0; a < com.length; a++)
            com[a].setEnabled(true);
        shuffleStart();
        reset.setEnabled(true);
        total = new TimeCalc();
    } else if (ae.getSource() == reset && (!doneFlag)) {
        reSet();
        totalTime = 0;
        reset.setEnabled(false);
    } else if (ae.getSource() == restart && (doneFlag)) {
        reStart();
        totalTime = 0;
        reset.setEnabled(true);
    } else {
        for (int i = 0; i < m; i++) {
            if (ae.getSource() == b[i] && (!clicked)) {
                b[i].setVisible(false);
                cLabel = b[i].getLabel();
                cl = i;
                clicked = !clicked;
            } else if (ae.getSource() == b[i] && (clicked)) {
                b[cl].setLabel(b[i].getLabel());
                b[cl].setVisible(true);
                b[i].setLabel("'" + cLabel);
                clicked = !clicked;
            }
        }
    }
}

```

```

        checkCorrect();
    }
}
}

public void checkCorrect() {
    int checkComl = 0;
    for (int i = 1; i < m; i++) {
        if (b[i].getLabel().equals(String.valueOf(i)))
            checkComl += 1;
    }
    if (checkComl == 8) {
        totalTime = total.getTimeInSeconds();
        for (int i = 0; i < m; i++)
            b[i].setVisible(false);
        doneFlag = true;
        restart.setEnabled(true);
        reset.setEnabled(false);
        msg = "Congratulations!, you Finished it in " + totalTime + " seconds !!";
        repaint();
    }
}

public void paint(Graphics g) {
    if (doneFlag) {
        setBackground(Color.BLACK);
        setForeground(Color.WHITE);
    } else
        setBackground(Color.WHITE);
    g.setFont(new Font("Serif", Font.PLAIN, 24));
    g.drawString(msg, 30, 250);
}

public void shuffleStart() {
    for (int i = 0; i < m; i++) {
        Random number = new Random();
        int num = number.nextInt(9);
        swap(num, i);
    }
}

public void reStart() {

```

```

        for (int i = 1; i < m; i++) {
            b[i].setVisible(true);
            b[i].setLabel(String.valueOf(i));
        }
        b[0].setVisible(true);
        b[0].setLabel("9");
        doneFlag = false;
        Component[] com = buttonPanel.getComponents();
        for (int a = 0; a < com.length; a++)
            com[a].setEnabled(false);
        restart.setEnabled(false);
        repaint();
    }

    public void reSet() {
        for (int i = 1; i < m; i++) {
            b[i].setLabel(String.valueOf(i));
        }
        b[0].setLabel("9");
        Component[] com = buttonPanel.getComponents();
        for (int a = 0; a < com.length; a++)
            com[a].setEnabled(false);
    }

    public void swap(int x, int y) {
        String temp = b[x].getLabel();
        b[x].setLabel(b[y].getLabel());
        b[y].setLabel(temp);
    }

    public static void main(String ar[]) {
        ButtonDrag cd = new ButtonDrag();
        cd.setSize(new Dimension(500, 500));
        cd.setTitle("Button Game");
        cd.setVisible(true);
    }
}

class MyWindowAdapter extends WindowAdapter {
    public void windowClosing(WindowEvent we) {
        System.exit(0);
    }
}

```

```
class TimeCalc {  
    private final long startedMillis = System.currentTimeMillis();  
  
    public int getTimeInSeconds() {  
        long nowMillis = System.currentTimeMillis();  
        return (int) ((nowMillis - this.startedMillis) / 1000);  
    }  
}
```

LAB 10

```
/* Rushi Hundiwala */
/* 1BM22CS224 */

public class DeadlockExample {

    public static void main(String[] args) {
        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 Object lock1 = new Object();
    private 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
        }
    }
}
```