LAB-1

```
import java.util.Scanner;

public class QuadraticSolver{
    public static void main(String args[])
    {
    Scanner sn = new Scanner(System.in);
    double a = scanner.nextDouble();

    System.out.println("Enter the coefficient
     b:");

    double b = sn.nextDouble();

    System.out.println("Enter the
     coefficient c:");

    double c= sn.nextDouble();

    double discriminant = b*b -4*a*c;

    if(discriminant >=0)
    {
    double root1 = (-b + Math.sqrt(discriminant))
                    /(2*a);
    double root2 = (-b - Math.sqrt(discriminant))
                    /(2*a);

    System.out.println("Real solutions:")
    System.out.println("Root 1:" + root1);
    System.out.println("Root 2:" + root2);
}
```

Date __/__/__
Page

```
else
{
  System.out.println("No real solutions.");
}
}
}
```

OUTPUT:

Enter coefficient a: 1
Enter coefficient b: 2
Enter coefficient c: 1

Real solutions:
Root 1: -1.0
Root 2: -1.0

Date 9/1/23
Page

Program 2 :

```
import java.util.Scanner;
class Subject {
    int subjectMarks;
    int credits;
    int grade;
}
class Student {
    Subject subjects[];
    String name, usn;
    double SGPA;
    Scanner s = new Scanner(System.in);
    Student() {   // constructor
        subjects = new subject[2];
        for (int i=0; i<2; i++) {
            subjects[i] = new Subject();
        }
    }
void getd() {
    System.out.println("Enter the student name");
    name = s.nextLine();
    System.out.println("Enter the usn:");
    usn = s.nextLine();
}
void getmarks() {
    for (int i= 0; i<2; i++) {
        System.out.println("Enter details
```

Scanned with CamScanner

```
for subject: " + (i+1));
System.out.println("Enter Marks:");
Subject[i].subjectMarks = S.nextInt();
System.out.println("Enter the credits");
Subject[i].credits = S.nextInt();

if(Subjects[i].subjectMarks >= 90){
        Subjects[i].grade = 10;
    }
else if
    (Subjects[i].subjet.Marks >= 80){
        Subjects[i].grade = 9;
    }
    .
    .
    .
} // Last if statement)
} // For loop
} // getMarks();

void getSGPA(){
    double totalcre=0, sum=0;

    for(int i=0; i<2; i++){
        totalcre += Subjects[i].credits;
        sum += (Subjects[i].grade *
                Subjects[i].credits);
    }
SGPA = sum / totalcre;
```

Date __/__/__
Page

```
void display(){

System.out.println("\n Result : \n
                    Name : \n"+name
                    +"USN : \n"+usn
                    +" SGPA : \n"+SGPA);
}
} // class student

class LAB2 {
public static void Main(){
    Student S1 = new Student();
    S1. getd();
    S1. getmarks();
    S1. getSGPA();
    S1. display();
}
}
```

9/11/24

LAB-3

```
import java.util.Scanner;
class Book {
    String name, author;
    int price, num_pages;
    Scanner in = new Scanner(System.in);
    Book(String name, String author,
         int price, int num_pages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

void getd()
{
    System.out.println("Enter the name of
                        the book:");
    name = in.nextLine();
    System.out.println("Enter the author:");
    author = in.nextLine();
    System.out.println("Enter price:");
    price = in.nextInt();
    System.out.println("Enter no. of pages:");
    num_pages = in.nextInt();
}

void display() {

    System.out.println("name :\n author:
                        \n price :\n no. of
                        pages:\n" + name
                        + author + price
                        + num_pages);
}
```

```
public String toString()
{
    return name + author + price +
        num_pages ;
}

class Lab3
{
  public static void main(String args[])
  {
     int n;
     book b[];
     Scanner in = new Scanner(System.in);
  System.out.println(" Enter the total no. of
     books");
  n = in.nextInt();
  b = new book[n];
  for(int i=0 ; i<n ; i++)
  {
System.out.println(" Enter details ");
String name = in.next();
String author = in.next();
int price = in.nextInt();
int num_pages = in.nextInt();
b[i] = new Book(name, author, price, num_pages);
}
for(int i= 0; i<n ; i++)
 {
Sys.tem.out.println(b[i].name + b[i].author +
                    b[i].price + b[i].num_pages);

 }
}
}
```

Date
Page

VARSHA

#impact

OUTPUT:

Enter the total no. of books:
2
Enter details for book 1:
Merchant of venice
shakespeare
1000
300
Eneter details for book 2:
If tmr never dies
Sidney sheldon
350
270

Book details 1:
Merchant of venice
shakespeare
1000
300
Book details 2:
If tmr never dies
Sidney sheldon
350
270

LAB-4:

```
import java.lang.Math;
import java.util.Scanner;
abstract class Shape{
    protected int int1, int2;
    abstract void printArea();
    Scanner in = new Scanner(System.in);
}
class Rectangle extends Shape{

    Rectangle()
    {
        System.out.println("Rectangle object
            created");
    }
    void printArea(){
    int area;
    System.out.println("Enter the length
    and Breadth of the rectangle:");
    int1 = in.nextInt();
    int2 = in.nextInt();
    area = int1 * int2;
    System.out.println("Area is:"+area);
}
class Triangle extends Shape{
    Triangle(){
        System.out.println("Triangle object
            is created");
    }
    void printArea(){
    double Area;
    System.out.println("Enter the base
    and height:");
```

```
Int1 = sn.nextInt();
Int2 = sn.nextInt();
area = 0.5 * Int1 * Int2;
System.out.println("Area of
  the Triangle :"+area);
}
}

class Lab4 {
  public static void main(String args[])
    {
      Rectangle obj1 = new Rectangle;
      obj1.printArea();
      Triangle obj2 = new Triangle();
      obj2.printArea();
    }
}
```

OUTPUT:

Rectangle object created
Enter the length and breadth:
3 4
Area is: 12
Triangle object created
Enter the base and height:
4 5
Area of the Triangle : 10.0

LAB - 5

Date __/__/__
Page _____

```
import java.util.Scanner;
class Account
{
    String customerName;
    long accountNumber;
    String accountType;
    double balance;
    Account(String customerName,
            long acount number,
            String accountType,
            double balance)
    {
        this.name = customerName;
        this.accnum = accountNumber;
        this.acctype = accountType;
        this.balance = balance;
    }

void deposit(double amount)
{ balance += amount;
  System.out.println("Deposit successful");
}

void display()
{
    System.out.println(account number +
                       accountType +
                       customerName +
                       balance);

}
class CurAcc extends Account
{
```

Date / /
Page

```
double minimum bal = 1000;
double service = 100;

public CurAcc (String c_name,
               String a_type,
               long acc_num,
               double balance)
{
super (customerName, accountNumber,
       account type = "current Account",
       balance);
}
class Savings extends Account{
double interestRate = 0.05;

public Savings (String ac_name,
                String a_type,
                long acc_num,
                double balance)
{
  super (customerName, accnombor,
         "savings Account", balance);
}
public class lab5 {
  public static void main (String args[]
  {

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

```
CurAcc obj1 = new CurAcc ("Ankit",
    1234567, 1500);
obj1.display();
obj1.Main();
System.out.println ("Enter the deposit
amt: ");

double dep = sn.nextDouble();
obj1.deposit(dep);

Savings obj2 = new Savings ("SS",
    45678910, 5000);
Obj2.display();

}
}
```

OUTPUT:

```
Account Number: 1234567
Customer Name: Ankit
Account type: Current Account
Current balance: 1500.0
Enter the deposit amount:
100
Deposit successful
Account number: 45678910
Customer name: SS
Acount Type: Savings Account
Current Balance: 5000.0
```

LAB-6

generics:

```
class stack <A, V> {

    A a, b, c, d, e;
    V f, g, h, p, j;

    void setA (A a, A b, A c, A d, A e){
        this.a = a;
        this.b = b;
        this.c = c;
        this.d = d;
        this.e = e;
    }

    void setV (V f, V g, V h, V p, V j)
    {

        this.f = f;
        this.g = g;
        this.h = h;
        this.p = p;
        this.j = j;
    }


class generics {
    public static void main (String
            args []) {

    Stack<Integer, Double = new Stack
    <Integer, Double> ();
    obj1.setA (1, 1, 1, 1, 1);
```

Date __/__/__
Page _____

```
obj1.setV(1.68, 1.69, 1.70, 1.71, 1.72);

System.out.println(obj1.a);
System.out.println(obj1.b);
System.out.println(obj1.c);
System.out.println(obj1.f);
System.out.println(obj1.g);
System.out.println(obj1.h);

    }
}
```

STRINGS

OUTPUT

1) Java
   V
   Java
   ABCDEF
   CDE

2) 3
   abc abc

3) Dimensions are 10.0 by 14.0 by 12.0
   Box b: Dimensions are 10.0 by 14.0 by 12.0

4) Bms

5) 72
   H

6) Bms equals Bms → true
   Bms equals cig → False
   Bms equals BMS → True
   Bms equals BMS → FALSE

7) substrings are matched

8&9) It is true
It is true

10) Hello.equals Hello → True
Hello == Hello → False

11) batch
apple
ball
cat
dog
ent
tree
gun
hen
pen
jug
kite
lift
man
net
orange
parrot
queen
ring
star
tree
umbrella

van
xmas
yatch
zee

12) 1
2
3
4
5
6
7
8
9
10

was

Date __/__/__
Page _____

13) Thwas a test. Thwas was too.
Thes was a test. Thwas was too.
This as a test. Thwas was too.
Thes as a test. Thes was too
This Ps a test. Thes Ps too.

14) Hello world

15) Original: I go to college
Replaced: I go to commege

16) Hello world

17) Enter detail for student 1:
Reg NO: 50
Full Name: Ankit.
Semester: 3
CGPA: 9.8

Similarly we enter details for 5 students

18) buffer before = Hello
Char At(1) before = e
buffer after = HP
Char At(1) after = P
H
e
e
HP 42
HPLLLe 42
24 ERPLPH

6/2/24

19) Eagle:
Eagle flies high on the sky
Eagle screeches loudly

Hawk:
Hawk soars gracefully through the air
Hawk emits a piercing cry.

generics OUTPUT:

1
1
1
1
1
1.6799
1.636
1.356
1.345
1.879

LAB 7:

Internal class:

```
package CIE;
import java.util.Scanner;
public class Internal extends Student{
    int i;
    int marks[] = new int[50];
    Scanner sc= new Scanner(System.in);
    public Internal(){
        System.out.println("Internal obj");
    }
    public void getId(){
        for(i=0; i<5; i++){
            System.out.println("Enter
the marks for Subject"+(i+1));
    marks[i] =sc.nextInt();
    }
}
}
```

```
package CIE;
import java.util.Scanner;
public class Student{
    protected String usn, name;
    protected int sem;
    Scanner sc = new Scanner(System.in);
    public Student(){
        System.out.println("Student
        Created\n");
    }
    public void getd(){
    System.out.println("Enter USN:");
    System.out.println("Enter Name:");
    System.out.println("Enter Sem:");
```

```
usn=pn.neltLine();
name = pn.nextLine();
sem = pn.nextInt();

public void display(){
    System.out.println("student
    Details" "Name:" +name);
    System.out.println("USN: \n"+usn);
    System.out.println("semester \n"+sem);
}
}

package SEE;
import CIE.Student;
import java.util.Scanner;
public class External extends Student{
    int p;
    int marksSEE[] = new int[10];
    Scanner pn = new Scanner(System.in);
    External(){
        System.out.println("External
        object is created \n");
    }

    void getd(){
        for(p=0; p<5; p++){
            System.out.println("Enter
            the marks for subject \n"+
            (p+1));
            marksSEE[p] = pn.nextInt();
        }
    }
}
```

Date __/__/__
Page _____

```
package Result;
import CIE.Internals;
import SEE.Externals;
public class FinalMarks {
    public static void main(String args[]){
    int n = 2;
    Internal i = new Internal[n];
    External e = new External[n];
for(int i=0; i<1 ; i++){
        i[i] = new Internal();
        i[i].getId();
        i[i] = getId();
        e[i] = new External();
        e[i] = getId();
        e[i] = geted();
    }
public static void calc(Internal i,
        External e){
    int[] final = new int[5];
    for(int i=0; i<5 ; i++){
        finalM[i] = i.iMarks[i] +
                    e.externalMarks[i];
    System.out.println("Course" + (i+1) +
                    ":" + finalM[i]);
    }
}
}
```

LAB - 8

```
import java.util.Scanner;

class WrongAgeException extends
    Exception {

    WrongAgeException (String m)
        {
            super (m);
        }
}

class Father {
    int fage;
    Father (int fage) throws WrongAge
    Exception {
    if (fage <=0) {
        throw new WrongAge Exception
        (" age !=0 );
    }
    else {
        this.fage = fage;
        System.out.println ("Father's Age
        is: "+fage);
    }
}

class Son extends Father {
    int sage;
    Son (int sage, int fage)
    throws WrongAge Exception {
        super (fage);
        if (sage >= fage)
        {
            throw new WrongAge Exception ("
```

```
        Son's age cant be >= Fathers age");
    }
    else
    {
        this.sage = sage;
        System.out.println("Son's age is:"
        + sage);
    }
}
}

public class Lab8 {
    public static void main (String args[])
    {
        Scanner sn = new Scanner (System.in);

        try {
            System.out.println("Enter father's
                                age:");
            int nf = sn.nextInt();
            Father fl = new Father (nf);
            System.out.println("Enter son's age:");
            int ns = sn.nextInt();
            Son s1 = new Son (ns);
        }
        catch (wrongAgeException e) {
            System.out.println("check age:" +e);
        }
    }
}
```

OUTPUT:

Enter the Father's age:
40
Father's age is: 40
Enter the son's age:
20
Father's age is: 40
Son's age is: 20.

30/1/24

Lab – 9

Date __/__/__
Page _____

```
public class lab9 {
    static class BMSThread extends
    Thread {
    public void run() {
        try {
            while (true) {
                System.out.println("BMS
            College of engineering");
            Thread.sleep(10000);
        }
        catch (Exception e) {
            System.out.println("Exception
            handled" +e);
        }
    }
}

static class CSEThread extends Thread{
    public void run() {
        try{
            while (true){
                System.out.println("CSE");
                Thread.sleep(2000);
            }
        }
        catch (Exception e)
            system.out.println("Exception
            handled" +e);
        }
    }
}
```

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

BMSThread bms = new BMSThread();
 bms.start();
CSEThread cse = new CSEThread();
 cse.start();
 }
}
```

OUTPUT:

```
BMS college of Engineering
CSE
CSE
CSE
CSE
CSE
BMS college g. Engineering
CSE
CSE
:
:
```

LAB-9 AWT

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo {
  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
    devider 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);
```

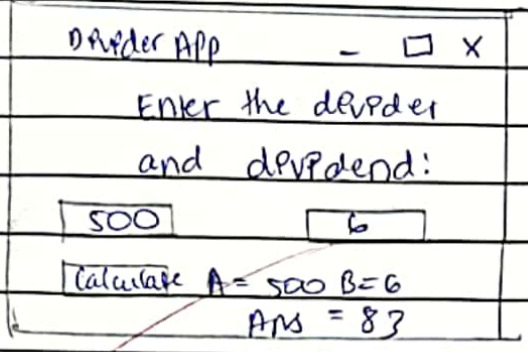
```
button.addActionListner(new ActionListner()
{
  public void actionPerformed(ActionEvent
    evt){
try{
    int a = Integer.parseInt(aftf.getText());
    int b = Integer.parseInt(bftf.getText());
    int ans = a/b;

err.setText("");
alab.setText("A =" +a);
blab.setText("B. =" + b);
anslab.setText("Ans =" +ans);
}
catch(NumberFormatException e){
    alab.setText("");
    blab.setText("");
    anslab.setText("");
    err.setText("B should be Non-zero!");
  }
});

frm.setVisible(true);
}

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

OUTPUT:



20/2/24

Functions used:

→ JFrame:
It contains the main window of the application

→ SetSize(int width, int height):
It is used to set the size of the GUI.

→ SetLayout:
Used to set the container.

→ JLabel:
Displays text or images

→ Addframe:
used to add a new frame.

LAB - 10 A

Date __/__/__
Page _____

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

synchronized void put(int n){
    while(valueSet)
        try {
            System.out.println("Producer
            waiting \n");
            wait();
        }
        catch (InterruptedException e){
            System.out.println("Exception
              handled");
        }
    this.n = n;
```

```
        valueSet = true;
        System.out.println("Put : " +n);
        System.out.println("InterruptedConsu
            -mer");
        notify();
    }
}

class Producer implements Runnable {
    Q q;

    Producer(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 Lab10a {
    public static void main(String
        args[])
```

```
{
    Q q = new Q();
    new Producer(q);
    new Consumer(q);
    System.out.println("Press Control-C to stop.");
  }
}
```

OUTPUT:

```
Put: 0
IntPmate customer
Producer waiting
Got: 0
IntPmate producer
Put: 1
IntPmate customer
Producer waiting
Consumed: 0
Got: 1
IntPmate Producer
Consumed: 1
```

Date __/__/__
Page

LAB-10 B

```
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();
}
  Sychronized 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("Inside B.last()");
```

```
    }
}

class Lab10b implements Runnable {
    A a = new A(); B b = new B();

Lab10b() {
    Thread.currentThread().setName
        ("Main Thread");
Thread t = new Thread(this, "Racing
    Thread");
t.start();
 a.foo(b);
System.out.println("Back in main T");
}
public void run(){
        b.bar(a);
        System.out.println("Back in other T");
}
public static void main(String args[])
        new Deadlock();
}
```

OUTPUT:

Racing Thread entered B.bar
MainThread entered A.FOO
MainThread trying to call B.last()
Racing thread trying to call A.last()

13/2/24

LAB PROGRAMS

LAB PROGRAM 1:

Q) 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.

A)

```
import java.util.Scanner;

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

        // Input coefficients a, b, and c
        System.out.println("Enter coefficient a:");
        double a = scanner.nextDouble();

        System.out.println("Enter coefficient b:");
        double b = scanner.nextDouble();

        System.out.println("Enter coefficient c:");
        double c = scanner.nextDouble();

        // Calculate discriminant
        double discriminant = b * b - 4 * a * c;

        // Check discriminant for real solutions
        if (discriminant >= 0) {
            // Calculate solutions using quadratic formula
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);

            // Display solutions
            System.out.println("Real Solutions:");
            System.out.println("Root 1: " + root1);
            System.out.println("Root 2: " + root2);
        } else {
            System.out.println("No real solutions. Discriminant is negative.");
        }
    }
}
```

LAB PROGRAM 2:

Q) 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.

A)

```
import java.util.Scanner;

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

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

   Student() {
      subjects = new Subject[2];
      for (int i = 0; i < 2; i++) {
         subjects[i] = new Subject();
      }
      s = new Scanner(System.in);
   }

   void getStudentDetails() {
      System.out.print("Enter Name: ");
      name = s.nextLine();
      System.out.print("Enter USN: ");
      usn = s.nextLine();
   }

   void getMarks() {
      for (int i = 0; i < 2; i++) {
         System.out.println("Enter details for Subject " + (i + 1));
         System.out.print("Enter Marks: ");
         subjects[i].subjectMarks = s.nextInt();
         System.out.print("Enter Credits: ");
         subjects[i].credits = s.nextInt();
```

```
            // Calculate grade based on marks
            if (subjects[i].subjectMarks >= 90) {
                subjects[i].grade = 10;
            } else if (subjects[i].subjectMarks >= 80) {
                subjects[i].grade = 9;
            } else if (subjects[i].subjectMarks >= 70) {
                subjects[i].grade = 8;
            } else if (subjects[i].subjectMarks >= 60) {
                subjects[i].grade = 7;
            } else if (subjects[i].subjectMarks >= 50) {
                subjects[i].grade = 6;
            } else if (subjects[i].subjectMarks >= 40) {
                subjects[i].grade = 5;
            } else {
                subjects[i].grade = 0; // Fail
            }
        }
    }

    void computeSGPA() {
        double totalCredits = 0;
        double weightedSum = 0;

        for (int i = 0; i < 2; i++) {
            totalCredits += subjects[i].credits;
            weightedSum += subjects[i].grade * subjects[i].credits;
        }

        SGPA = weightedSum / totalCredits;
    }

    void displayResult() {
        System.out.println("\nResult:");
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}

class LAB2 {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
```

```
        s1.computeSGPA();
        s1.displayResult();
    }
}
```

—----------

LAB PROGRAM 3:

Q) 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.

A)

```
import java.util.Scanner;
class Book{
    String name,author;
    int price,num_pages;
    Scanner in=new Scanner(System.in);
    Book(String name,String author,int price, int num_pages){
        this.name=name;
        this.author=author;
        this.price=price;
        this.num_pages=num_pages;

    }
    void getd(){
        System.out.println("enter the name of the book:");
        name=in.nextLine();
        System.out.println("enter the name of the author:");
        author=in.nextLine();
        System.out.println("enter the price of the book");
        price=in.nextInt();
        System.out.println("enter the number of pages in the book:");
        num_pages=in.nextInt();
    }
    void display(){
        System.out.println("name of the book:"+ name);
        System.out.println("name of the author:"+ author);
        System.out.println("price of the book:"+ price);
        System.out.println("no. of pages in the book:"+ num_pages);
    }
    public String toString(){
        String name,author,price,num_pages;
```

```
        name="book name:"+this.name+"\n";
        author="author name:"+this.author+"\n";
        price="book price:"+this.price+"\n";
        num_pages="number of pages in the book"+this.num_pages+"\n";
        return name+author+price+num_pages;

    }
}
public class lab3{
    public static void main(String args[]){
        int n;
        Book b[];
        Scanner in=new Scanner(System.in);
        System.out.println("enter the total number of books:");
        n=in.nextInt();
        b=new Book[n];
        for(int i=0;i<n;i++){
            System.out.println("\nEnter details for Book " + (i + 1) + ":");
            System.out.println("Name: ");
            String name = in.next();
            System.out.println("Author: ");
            String author = in.next();
            System.out.println("Price: inr");
            int price = in.nextInt();
            System.out.println("Number of Pages: ");
            int num_pages = in.nextInt();
            b[i]=new Book(name,author,price,num_pages);
        }
        System.out.println("\n\n\n");
        for(int i=0;i<n;i++){
            System.out.println("Book details:"+(i+1));
            System.out.println("book name:"+b[i].name);
            System.out.println("author name:"+b[i].author);
            System.out.println("book price:"+b[i].price);
            System.out.println("number of pages in the book:"+b[i].num_pages);
            System.out.println("\n");

    }
}
}
```

LAB PROGRAM 4:

Q) 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 contains only the method printArea( ) that prints the area of the given shape.

A)

```
import java.lang.Math;
import java.util.Scanner;
abstract class Shape{
   protected int int1,int2;
   abstract void printArea();
   Scanner in=new Scanner(System.in);
}
class Rectangle extends Shape{
   Rectangle(){
      System.out.println("Rectangle object created");
   }
   void printArea(){
   int area;
   System.out.println("Enter the length and breadth of the Rectangle:");
   int1=in.nextInt();
   int2=in.nextInt();
   area=int1*int2;
   System.out.println("Area of the Rectangle:"+area);
}
}
class Triangle extends Shape{
   Triangle(){
      System.out.println("Triangle object created");
   }
   void printArea(){
   double area;
   System.out.println("Enter the base and height of the triangle:");
   int1=in.nextInt();
   int2=in.nextInt();
   area=0.5*int1*int2;
   System.out.println("Area of the triangle:"+area);
}
}
class Circle extends Shape{
   Circle(){
      System.out.println("Circle object created");
   }
```

```
    void printArea(){
    double area;
    System.out.println("Enter the Radius of the Circle:");
    int1=in.nextInt();
    area=Math.PI*int1*int1;
    System.out.println("Area of the triangle:"+area);
}
}
class lab4{
    public static void main(String[] args) {
        Rectangle obj1=new Rectangle();
        obj1.printArea();
        Triangle obj2=new Triangle();
        obj2.printArea();
        Circle obj3=new Circle();
        obj3.printArea();
    }
}
```

—----------

LAB PROGRAM 5:

Q) Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called a 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 a check 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.

A)

```
import java.util.Scanner;

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

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

```
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit successful. Updated balance: " + balance);
    }

    public void displayBalance() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Current Balance: " + balance);
    }
}

class CurAcct extends Account {
    double minimumBalance = 1000; // Minimum balance required for a current account
    double serviceCharge = 100; // Service charge imposed if balance falls below the minimum

    public CurAcct(String customerName, long accountNumber, double balance) {
        super(customerName, accountNumber, "Current Account", balance);
    }

    public void checkMinimumBalance() {
        if (balance < minimumBalance) {
            balance -= serviceCharge;
            System.out.println("Service charge of " + serviceCharge + " imposed for falling below minimum
balance.");
            System.out.println("Updated balance: " + balance);
        }
    }
}

class SavAcct extends Account {
    double interestRate = 0.05; // Compound interest rate for savings account

    public SavAcct(String customerName, long accountNumber, double balance) {
        super(customerName, accountNumber, "Savings Account", balance);
    }

    public void computeAndDepositInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest computed and deposited. Updated balance: " + balance);
    }
```

```
    public void withdraw(double amount) {
        if (balance >= amount) {
            balance -= amount;
            System.out.println("Withdrawal successful. Updated balance: " + balance);
        } else {
            System.out.println("Insufficient funds for withdrawal.");
        }
    }
}

public class lab5{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Example Usage
        CurAcct currentAccount = new CurAcct("John Doe", 123456789, 1500);
        currentAccount.displayBalance();
        currentAccount.checkMinimumBalance(); // No service charge

        System.out.println("\nEnter the deposit amount: ");
        double depositAmount = scanner.nextDouble();
        currentAccount.deposit(depositAmount);
        currentAccount.checkMinimumBalance(); // Service charge imposed

        SavAcct savingsAccount = new SavAcct("Jane Doe", 987654321, 5000);
        savingsAccount.displayBalance();

        System.out.println("\nEnter the withdrawal amount: ");
        double withdrawalAmount = scanner.nextDouble();
        savingsAccount.withdraw(withdrawalAmount);
        savingsAccount.computeAndDepositInterest();
        savingsAccount.displayBalance();
    }
}
```

LAB PROGRAM 6:

Q) Create a package CIE which has two classes- Student and Internals. The class Students have 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.

A)

```
package CIE;

import java.util.Scanner;

public class Student {
   protected String usn, name;
   protected int sem;

   public Student() {
      System.out.println("STUDENT OBJECT CREATED");
   }

   public void getDetails() {
      Scanner in = new Scanner(System.in);
      System.out.println("Enter the student USN, Name, and Semester:");
      usn = in.nextLine();
      name = in.nextLine();
      sem = in.nextInt();
   }

   public void displayDetails() {
      System.out.println("STUDENT DETAILS:\nName: " + name + "\nUSN: " + usn + "\nSemester: " +
sem);
   }
}

package CIE;
import CIE.Student;
import java.util.Scanner;

public class Internal extends Student {
   public int[] internalMarks = new int[5];

   public Internal() {
      System.out.println("INTERNALS OBJECT CREATED");
```

```
    }

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

package SEE;

import CIE.Student;
import java.util.Scanner;

public class External extends Student {
    public int[] externalMarks = new int[5];

    public External() {
        System.out.println("EXTERNAL OBJECT CREATED");
    }

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

package Result;

import CIE.Internal;
import SEE.External;

public class FinalMarksDeclaration {
    public static void main(String[] args) {
        int n = 2; // Number of students, you can change this

        Internal[] internalsArray = new Internal[n];
```

```
        External[] externalsArray = new External[n];

        // Create objects and input details for Internals and Externals
        for (int i = 0; i < 1; i++) {
            internalsArray[i] = new Internal();
            internalsArray[i].getDetails();
            internalsArray[i].getInternalMarks();

            externalsArray[i] = new External();
            externalsArray[i].getDetails();
            externalsArray[i].getExternalMarks();
        }

        // Display details and calculate final marks for each student
        for (int i = 0; i < 1; i++) {
            System.out.println("\nDetails for Student " + (i + 1));
            internalsArray[i].displayDetails();
            externalsArray[i].displayDetails();
            calculateFinalMarks(internalsArray[i], externalsArray[i]);
        }
    }

    private static void calculateFinalMarks(Internal internals, External external) {
        int[] finalMarks = new int[5];

        System.out.println("\nFinal Marks for Student:");
        for (int i = 0; i < 5; i++) {
            // Assuming final marks are a combination of internal and external marks
            finalMarks[i] = internals.internalMarks[i] + external.externalMarks[i];
            System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}
```

LAB PROGRAM 7:

Q) Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and a 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.

A)

```
import java.util.Scanner;

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


class Father {
   int fage;
   Father(int fage) throws WrongAgeException {
      if (fage <= 0) {
         throw new WrongAgeException("Father's age cannot be negative or zero");
      } else {
         this.fage = fage;
         System.out.println("Father's age is: " + fage);
      }
   }
}

class Son extends Father {
   int sage;

   Son(int sage, int fage) throws WrongAgeException {
      super(fage);
      if (sage >= fage) {
         throw new WrongAgeException("Son's age cannot be greater than or equal to father's age");
      } else if (sage <= 0) {
         throw new WrongAgeException("Son's age cannot be negative or zero");
      } else {
         this.sage = sage;
         System.out.println("Son's age is: " + sage);
      }
   }
}
```

```
public class lab8 {
   public static void main(String[] args) {
      Scanner in = new Scanner(System.in);

      try {
         System.out.println("Enter the father's age:");
         int nf = in.nextInt();
         Father f1 = new Father(nf);

         System.out.println("Enter the Son's age:");
         int ns = in.nextInt();
         Son s1 = new Son(ns, nf);
      } catch (WrongAgeException e) {
         System.out.println("Check age: " + e);
      }
   }}
```

LAB PROGRAM 8:

Q) 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.

A)

```
public class lab9 {
   static class BMSThread extends Thread {

      public void run() {
         try {
            while (true) {
               System.out.println("BMS College of Engineering");
               Thread.sleep(10000);
            }
         } catch (InterruptedException e) {
            System.out.println("Exception handled"+e);
         }
      }
   }

   static class CSEThread extends Thread {
      public void run() {
         try {
            while (true) {
               System.out.println("CSE");
               Thread.sleep(2000);
            }
         } catch (InterruptedException e) {
            System.out.println("Exception handled"+e);
         }
      }
   }

   public static void main(String[] args) {

      BMSThread bmsThread = new BMSThread();
      bmsThread.start();
      CSEThread cseThread = new CSEThread();
      cseThread.start();
   }
}
```

LAB PROGRAM 9:

Q) 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 is 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.

A)

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

class SwingDemo {
   SwingDemo() {
      // create jframe container
      JFrame jfrm = new JFrame("Divider App");
      jfrm.setSize(275, 150);
      jfrm.setLayout(new FlowLayout());

      jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

      // add text field for both numbers
      JTextField ajtf = new JTextField(8);
      JTextField bjtf = new JTextField(8);

      // calc button
      JButton button = new JButton("Calculate");

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

      // add in order :)
      jfrm.add(err);  // to display error
      jfrm.add(jlab);
      jfrm.add(ajtf);
      jfrm.add(bjtf);
      jfrm.add(button);
      jfrm.add(alab);
```

```
        jfrm.add(blab);
        jfrm.add(anslab);

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

                    // Reset error labels
                    err.setText("");
                    alab.setText("A = " + a);
                    blab.setText("B = " + b);
                    anslab.setText("Ans = " + ans);
                } catch (NumberFormatException e) {
                    // Clear other labels if one is invalid
                    alab.setText("");
                    blab.setText("");
                    anslab.setText("");
                    err.setText("Enter Only Integers!");
                } catch (ArithmeticException e) {
                    // Clear other labels if one is invalid
                    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 PROGRAM 10 (a):

Q) Demonstrate Inter process Communication

A)

```
class Q {
   int n;
   boolean valueSet = false;

   synchronized int get() {
      while(!valueSet)
         try {
            System.out.println("\nConsumer waiting\n");
            wait();
         } catch(InterruptedException e) {
            System.out.println("InterruptedException caught");
         }

      System.out.println("Got: " + n);
      valueSet = false;
      System.out.println("\nIntimate Producer\n");
      notify();
      return n;
   }

   synchronized void put(int n) {
      while(valueSet)
         try {
            System.out.println("\nProducer waiting\n");
            wait();
         } catch(InterruptedException e) {
            System.out.println("InterruptedException caught");
         }

      this.n = n;
      valueSet = true;
      System.out.println("Put: " + n);
      System.out.println("\nIntimate 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 lab10a {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}
```

LAB PROGRAM 10 (b):

Q) Demonstrate Deadlock

A)

```
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();
   }

   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();
   }

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

class lab10b implements Runnable {
   A a = new A();
   B b = new B();
```

```
    lab10b() {
        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 main thread");
    }

    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();
    }
}
```