B.M.S COLLEGE OF ENGINEERING BENGALURU

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfilment of the requirements for Lab

Bachelor of Engineering

in

Computer Science and Engineering

Submitted by:

REVANTH K (1BM22CS220)

Department of Computer Science and Engineering, B.M.S

College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019 2023-2024.

INDEX

Revanth K
1BM22CS220

①

```
Class helloWorld
{
    public static void main (String [] args)
    {
    System.out.println ("helloWorld");
    }
}
```

O/P :- helloWorld

②

```
Class rectangle Area {
        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 ="+ length);
        System.out.println (" breadth =" + breadth);
        System.out.println ("Area=" + Area);
    }
}
```

O/P :-

```
length = 5
breadth = 5
Area = 25
```

Revanth K
1BM22CS220

③ Array :-

```
class AutoArray {
public static void main (String args[])
{
int month_days ={31, 28, 31, 30, 31, 30, 31, 31, 30, 31};
System.Out.println ("April has "+ month_days [3] + "days.");
 }
}
```

O/P → April has 30 days.

④.

```
import java.util Scanner;
class scanner {
public static void main (String args[])
{
 int a; float b ; String s;
 Scanner in = new Scanner (System.in);
 System.out.println ("enter a string :");
 s = in.nextLine();
System.out.println (" you entered string "+s);
System.Out.println (" enter an integer");
a = in.nextInt();
System.out.println ( "you entered integer "+a);
System.out.println ("enter a float");
b = in nextFloat ();
System.out.println ("you entered float"+b);
 }
```

Revanth K
1BM22CS220

O/p:-

```
enter a string:
revanth
you entered string revanth
enter an integer
3.
you entered integer 3
-enter a float
3.5
you entered float 2.5
```

(5)

```
import java.util.Scanner;
Class palendrome
{
  public static void main (String args[])
  {
    int n, t, rem, rev=0;
    Scanner SC = new Scanner (System.in);
    System.out.println ("enter a 5 digit number: ");
    n=SC.nextInt();
    t=n;
    while (t>0)
  {
    rem = t %10;
    rev = rev * 10 + rem;
    t=t/10;
  }
  if (rev == n)
  {
    system.Out.println ("Palindrome");
  }
```

Revanth K
1BM22CS220

```
else
{
    System.out.println ("not palindrome");
  }
 }
}
```

Output :-

```
Enter a 5 digit number:
 12321
Palindrome

Enter a 5 digit number
12345
not palindrome
```

(6) Quadratic :-

O/P :-

Output 1:

```
Enter the coefficients of a,b,c
1
-5
2
roots are real and distinct
root1 = 4.5615528128      root 2 = 4.5615528128
```

Kaurnth K
1BM22CS220

Output 2 :- enter the coefficients of a, b, c

```
1 2 1
roots are real and equal
root 1 = root 2 = -1.0
```

Output 3 :-

```
enter the coefficients of a,b,c
0 4 5
not a quadratic equation
enter a non zero value for a ;
1
roots are imaginary
root 1= -2.0 + i NaN
root 2= -2.0 - i NaN
```

(7)

```
import . java.util . Scanner ;
class fact {
public static void main ( String args [])
{
 int fac = 1 ;
System. out. println ("enter a number : ");
Scanner sc = new Scanner (System.in);
int n = sc. nextInt();
for (int i = 1 ; i<=n ; i++) {
 fac = fac * i ;
}
System.out.println ("the factorial : "+fac);
```

Revanth K
1BM22CS220

```
}
}
```

Output :-

```
Enter a number:
    5
The factorial is :
    120
```

⑧

```
import java.util.*;
class digits {
    public static void main (String args[]){
        long number, sum;
        Scanner sc= new Scanner (System.in);
        System.out.println ("Enter a 5-digit number:");
        number =sc.nextLong();
        for (sum=0;  number !=0 ; number = number /10){
                Sum=sum + number % 10;
        }
        System.out.println ("Sum of digits: "+sum);
    }
}
```

Output :-

```
Enter a 5-digit number:
  12345
Sum of digits : 15
```

[illegible] 10-12-22

Complete all

9)
```
import java.util.*;
class isprime
{
 Static void isprime (int n)
 {
    int i, m=0, flag=0;
    m=n/2
    if (n==0 ((n==1))
    {
     system.out.println(n+" is not a prime no");
    }
 }
else
 { for (i=2; i<=m; i++)
 {  ~~System.out.println(n+"is a prime number");~~
       if (n%i ==0)
   {
     System.out.println (n+" is not a prime number");
     flag = 1;
     break;
   }
   if (flag ==0)
   {
     System.out.println (n+"is a prime number");
   }
}  }
public static void main (String args[])
{
    int i;
    Scanner s = new Scanner (System.in);
    System.out.println ("Enter the value of i: ");
      i = s. nextInt ();
      isprime [i];
 }
}
```

<u>Output:-</u>

```
Enter the Value of i
  7
It is a prime number
```

[illegible]

SGPA Lab-2 :-

```
import java.util.Scanner;
Class Subject
{
    int subject Marks;
    int credits;
    String grade;
}
Class Student
{
   String name;
   String usn;
   double SGPA;
   Scanner s;
   Subject Subject[];
   Student()
   {
     int i;
     subject = new Subject[9];
     for (i=0; i<9; i++)
          Subject [i] = new Subject ();
       S = new Scanner (System.In);
   }
 void getStudent Details()
 {
    System.out.println("enter your name: ");
    name = S.next line ();
    System.out.println("enter your USN:");
    USN = S.next Line ();
 }
```

```
void getMarks ()
{
    int i;
  for (i=0; i<8 ; i++)
{
 System.out.println ("enter the marks and credits for course "+i+": ");
 System.out.println ( "marks:");
 int marks = S.nextInt();
System.out.println ("credits:");
int credit =S.nextInt();
Subject [i]. subject Marks= marks;
Subject [i]. credits = credit;
 if (marks >=0 90 && mark <=100)
{
      subject [i]. grade = "O";
 }
 else if (marks >= 80 && marks <90)
{
    subject [i]. grade ="A+";
 }
else if (marks >=70 && marks<80)
{
     subject (i).grade="A";
}
else if (marks >=66 && marks<70)
{
   subject [i].grade = "B+";
}
else if (marks >=50 && marks <60)
 {
```

```
}
void
{
   i,
dou
dou
doub
for (
{
total
Switch
{
 Case '
Case "A+
Case "A"
      k
case "B+":
        subject [i].grade = "B";
    }
    else if (marks >=40 && marks <50)
    {   subject [i].grade = "C";
    }
    else if (marks >=0 && marks <40)
    {
        subject.[i].grade = "F";
    }
  }
}
void compute SGPA()
{
   int i;
  double sgpa;
  double total Credits =0;
  double totalgrade points = 0;

  for (i=0; i<8; i++)
 {
   total credits + = subject [i].credits;
   switch (subject [i].grade)
   {
      case "O" : totalgrad points + = 10 * subject [i].credits;
                 break;
      case "A+" : total grade points+ = 9 * subject[i].credits;
                 break;
      case "A" : total grade points + = 8 * subject [i].credits;
                 break;
   case "B+" : total grade points + = 7 * subject [i].credits;
                 break;
```

```
Case "B": totalgradepoint+= 6 * subject [i]. Credits ;
 break ;
Case "C" : totalgrade points + = 5 * Subject [i]. credits;
 break ;
Case "F" : total grade points + = 0 * subject [i]. credits;
 break ;
 }
}
 sgpa = total. grade points / total credits ;
 System. out. println (" the sgpa is : " + sgpa);
 }
}
Class sgpa
{
   public static void main (String args[])
   {
      Student S1 = new Student ;
      S1.get Student Details ();
      S1.get Marks();
      S1.compute SGPA();
   }
}
```

Output :-

```
Enter your name :
Revanth K
Enter your USN
1BM22CS220
```

```
Enter the marks and credit for cours 0:
marks :
95
Credits :
4
Enter the marks and credits for cours 1:
marks :
92
credits
4

Enter the marks and credits for cours2:
marks :
90
credits :
4
Enter the marks and Credit for cours 3:
marks :
95
Credits :
4
Enter the mark for cour 4;
maks:
91
Credits
3
Enter the maks and cour 5:
Marks: 91
Credits = 4
```

```
Enter the marks and credits for course 7:
Mark : 90
credits :
  1

the sgpa is : 10.000
```

[illegible] 19.12.22

Lab-3

26/12/23

```
import java.util.Scanner;
class Books
{
   String name;
   String author;
   int price;
   Int num Pages;
   Books (String name, String author, int price, int num Pages)
{
      this.name = name;
      this.author = author;
      this.price = price;
      this.num Pages = numPages;
}
public String to String ()
{
        String name, author, price, numPages;
        name = " Book name: "+ this.name +"\n";
        author =" Author name:" + this.author + "\n";
        price = "Price :"+ this.price + "\n";
        numPages = " Number of pages :" + this.numPages + "\n";
    return name+ author + price+ numPages;
   }
}
public class Mainbook
{
  public static void main (String args [])
   {
       Scanner s = new Scanner (System.in);
       int n;
      int  i;
       String name;
       String author;
       int price;
      Int numPages;
   System.out.println ("Enter the no of books : ");
   n = s.nextInt();
   Books b[];
   b = new Books [n];
for (i=0; i<n; i++)
{
   system.out.println (" Details of book"+(i+1) + ":");
  System.out.println (" name of book :");
  name =s.next();
  system. out.println("author name :");
 author = s.next();
System.out.println("price :");
price = s.nextInt();
System.out.println("no of pages :");
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(b[i]);
}
}
}
```

Output:-

```
Enter the no of books :
2
Enter the details of book 1 :
Enter the name of book :
harry potter
Enter the author name :
jkrowlings
Enter the price:
1000
Number of pages:
800
```

```
Enter the details of book2:
Enter the name of book:
Revolutionary 2020

author name:
Chethan Bagath

price:

2000

number of pages:

1500

Book details:
Book name: harrypotter
Author name: j k Rowlings
Price: 1000
number of pages: 800


Book name: Revolutionay 2020
Author name: Chethan Bagath
Price: 2000
number of pages = 1500
```

[illegible] 26.12.23

Lab-4

21/1/24

```
import java.util.scanner;
class input Scanner
{
    protected Scanner scanner;
    public input Scanner()
    {
        scanner = new Scanner(System.in);
    }
}
abstract class shape extends input scanner
{
    double a,b;
    public shape()
    {
        super();
        System.out.println("the area of a:");
        a = scanner.next Double();
        System.out.println(" the area of b:");
        b=scanner.next Double();
    }
}
class rectangle extends shape
{
    ~~public rectangle()~~
    void area()
    {
        double area = a * b;
        system.out.println(" the area of rectangle is:" + area);
    }
}
```

```
class triangle extends shape
{
    [illegible]
    void area()
    {
     double area = 0.5*a *b;
     System.out. println ("the area of triangle is: "+area);
    }
}
class circle extends shape
{
    void area ()
    {
     double area = 3.14*a *a ;
     System.out . println (" the area of rectangle is: "+area);
    }
}
public Class mainArea
{
    public static void main (String []args)
    {
        rectangle r = new rectangle ();
        triangle t = new triangle ();
        circle c = new circle ();
        r.area ();
        t. area ();
        c. area ();
    }
}
```

output :

```
the area of a :
 2
the area of b :
 3
the area of a :
 4
the area of b :
 5
the area of a :
 6
the area of b :
 7
the area of rectangle is : 6.0
the area of triangle is : 10.0
the area of circle is : 113.039
```

[illegible]
2/1/24

Lab - 3 :-   9/01/24

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

```
}
class SavAcct extends account
{
    private static double rate =5;
    savAcct (String name, int acno, double balance)
    {
        super (name, acno, "Savings", balance);
    }
    void intrest ()
    {
        balance += balance*(rate)/100;
        System.out.println ("balance:" + balance);
    }
}
class curAcct extends account
{
    private double minbal =500;
    private double serviceCharge =50;
    curAcct (String name, int acno, double balance)
    {
        super (name, acno, "curent", balance);
    }
    void checkmin()
    {
        if (balance < minBal)
        {
            System.out.println ("balance is less than min balance, service
                                 charge imposed:" + service charge);
```

```
balance = service charges;
System.out.println ("balance is:" + balance);
}
}
}
class accountMain
{
public static void main (String a[])
{
Scanner = 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 accn = S.nextInt();
System.out.println ("enter the initial balance:");
double balance = S.nextDouble();
int ch;
account acc = new acc (name, acc no, type, balance);
Sav Acct sa = new SavAcct (name, accno, balance);
Cur Acct ca = new CurAcct (name, accno, balance);
while (true)
{
if (acc.type.equals ("savings"))
{
System.out.println ("\n Menu \n 1. 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 amnt : ");
            amount 1 = S.nextInt();
            Sa.deposit (amnt);
            break;

    Case 2: System.out.println ("enter the amt");
            amont 2 = S.nextInt ();
            Sa.withdraw (amt 2);
            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 ("\n Menu \n 1. deposit 2. withdraw 3 display
    System.out.println ("enter the choice");
    ch=S.nextInt ();
    Switch (ch)
    {
```

```
case 1: System.out.println ("enter the amt :");
        amnt 1 = s.nextInt();
        ca.deposit (amnt 1);
        break;

case 2: System.out.println ("enter the amt" );
        amnt 2 = s.nextInt();
        ca. withdraw (amt 2);
        ca. checkmin ();
        break;

case 3: ca.display ();
        break;

case 4: system.exit(0);

default : System.out.println ("invalid input");
    }
  }
}

}
}
```

→ Output:-

```
enter the name : revanth

enter the account number:
    2201

enter the initial balance:
    5000

menu
1. deposit  2. withdraw  3. display
```

```
entu choice:
2
entu amt:
600
Menu
1.deposit. 2.withdraw 3.display
ente the choice
3
name: Pavant acno:2201 type: curnt balan 5600
```

[illegible] 09.01.24

Lab-6 :-   16/01/24

Question 1 :-

type 1: BMSCE
type 2: BMSCE
type 3: BMSCE
type 4: MS
type 5: abcd

Question 2 :-

length of S1 = 5
concatenation of S1 and S2: BMSCEBMSCE

question 3 :-

toString(): 10

Question 5 :-

~~65,66,67,68,69,70.~~
65 66 67 68 69 70
B M S C E

Question 4 :-

The given string is: Welcome to bmsce college
The SrcBegin, SrcEnd, and dstBegin or values are: 11, 16 and 0
The value of character array: [b, m, s, c, e, ...]

Question 6 :-

BMSCE equals BMSCE → true

BMSCE equals College → false

BMSCE equals ignorecase BMSCE → true

Question 7 :-

Substring S matched

S1 = "BMSCE college";

S2 = "welcome to BMSCE college of engineering";

Question 8 :-

true

false

Question 9 :-

false

True

Question 10 :-

Hello equal Hello ⇒ True

Hello == Hello ⇒ False

Question 11 :-

The names in alphabetical order are:

apple
ball
cat
venn
watch

Question ~~11~~ 12 :-

Sorted numbers : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Question 13 :-

Thwas was a test thwas was, too

Question 14 :-

hello world

Question 15 :-

conmege

[illegible] 16.01.24

Lab-7 23/01/23

```
//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 input Student Details () {
    Scanner scanner = new Scanner (System.in);
    System.out.println("Enter USN: ");
    USN = scanner.next();
    System.out.print("Enter Name: ");
    name = Scanner.next();
    System.out.print("Enter Semester: ");
    Sem = Scanner.next Int();
}
public void display Student Details () {
    System.out.println("USN: "+ usn);
    System.out.println("Name: "+name);
    System.out.println("Semester: "+sem);
  }
}
```

23/01/23

```
* Internals.java
package CIE;
import java.util.Scanner;
public Class internals extends Student {
protected int marks [] = new int [5];
public void input CIE marks() {
    Scanner scanner = new Scanner (System.in);
    System.out.println("Enter the internal marks for "+ name);
    for (int i=0 ; i<5; i++) {
        System.out.print("Subject "+ (i+1) + " marks :");
        marks [i] = scanner. nextInt();
    }
  }
}

// Externals. java
package SEE;
import CIE.Internals;
import java.util.Scanner;
public Class Externals extends Internals {
    protected int marks [];
protected int final Marks [];

public Externals () {
    marks = new int [5];
    final Marks = new int[5];
}
public void inputSEEmarks () {
   Scanner Scanner = new Scanner (System.in);
   System.out.println ("Enter SEE Marks for"+name);
   for (int i=0 ; i<5 ; i++) {
     System.out.print (" Subject "+ (i+1) + " marks :");
     marks [i] = scanner.nextInt ();
   }
}

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

public void displayFinalMarks() {
   display Student Details ();
   for (int i=0; i<5 ; i++)
     System.out.println ("Subject" +(i+1)+": "+ finalMarks [i]);
}
}
```

```
//Main.java
import SEE.Ext
public Class Ma
  public static
    int n
    .Exte
   for (int
     f
     f
     S
     f
   }
  System
  for(
  }
}
```

```
//Main.java
import SEE.Externals;

public Class Main {
  public static void main (String args [J]) {

      int numOf Students = 2;
      Externals finalMarks [] = new Externals [num of students];

    for (int i=0 ; i<num of students ; i++) {
        final Marks[i] = new Externds();
        final Marks [i]. input Student Details();
        System.out.println ("Enter CIE marks");
        finalMarks[i]. input CIIE marks();
        System.out.println(" Enter SEE marks");
        finalMarks [i]. input SEE marks();

    }

System.out.println (" Displaying data : \n");
for(int i = 0; i< num of students ; i++){
        final Marks[i]. CalculateFinal Marks ();
        finalMarks [i]. display Final Marks ();
  }
 }
}
```

Output :-

```
Enter USN: 1BM22CS201
Enter Name : A
Enter Semester : 2
Enter CIE marks .
Enter Internal mark for A
Sub 1 : 45
Sub 2 : 46
Sub 3 : 42
Sub 4 : 43
Sub 5 : 49
Enter for SEE marks
Enter SEE marks for A
Sub 1 : 85
Sub 2 : 89
Sub 3 : 87
Sub 4 : 88
Sub 5 : 87
Enter USN : 1BM22CS220
Enter Name = B
Enter Semester : 2
Enter CIE marks
Enter Internal Marks for B
```

```
Sub 1 = 45

Sub 2 : 47

Sub 3 : 40

Sub 4 : 43

Sub 5 : 44

Enter SEE marks

Enter .SEE marks for B

Sub 1 : 90

Sub 2 : 78

Sub 3 : 79

Sub 4 : 89

Sub 5 : 80

Displaying data :

USN : 1BM22CS201

Name : A

Semester : 2

Sub 1 : 87

Sub 2 : 90

Sub 3 : 85

Sub 4 : 87

Sub. 5 : 87
```

[illegible]

USN : 1BM22CS220

Name : B

Semester : 2

Sub 1 : 90

Sub 2 : 86

Sub 3 : 76

Sub 4 : 88

Sub 5 : 75

[illegible]
23.01.24

Lab-8 :- 30/01/24

```
import java.util.Scanner;
class Wrong Age extends Exception
{
 public Wrong Age (String message)
 {
   Super (message);
 }
}
Class InputScanner
{
 protected Scanner S;
 public Input Scanner ()
 {
   S = new Scanner (System.in);
 }
}
class Father extends InputScanner
{
 protected int father Age;
 public Father () throws WrongAge
 {
  System.out.println ("Enter Father's Age: ");
  father Age = S.next Int();
  if (father Age<0)
  {
  throw new WrongAge ("Age cannot be negative: ");
 }
}
public void display ()
  {
    system.out.println ("Father's Age :"+ father Age);
  }
}

Class Son extends Father
{
   private int son Age;
   public Son () throws Wrong Age
   {
     super ());
     System.out.println ("Enter Son's Age :");
     Son Age=S. nextInt ();
     if (Son Age > father Age)
     {
       throw new wrongAge (" Son's Age can't be greater than
                                  father's age ");
     }
   else if (son Age<0)
   {
     throw new Wrong Age ("Age can't be negative");
   }
  }
 public void display ()
 {
   super.display ();
   System.out. println ("Son's Age :"+ Son Age);
 }
}
```

```
public class FatherSonAge
{
  public static void main (String args[])
  {
    try
    {
      son Son =new Son();
      Son.display();
    }
    Catch (wrong Age)
    {
      System.out.println("Error: "+ e.getMessage());
    }
  }
}
```

Output:-

```
Enter Father's Age:
 50
Enter Son's Age:
 25
Father's Age:50
Son's Age:25
Enter Father's Age:
 10
Enter Son's Age:
 20
Error: Son's age cannot be greater than father's age
Enter Father's Age:
 -1
Error: Age cannot be negative
```

[illegible] 30.01.24

Lab - 9 :-

```
Class BMSThread extends Thread {
  @ override
  public void run() {
     while (true) {
              System.out.println ("BMS College of Engineering");
       try {
          Thread.sleep (10000);
       } Catch (InterruptedException e) {
            e.printStackTrace ();
       }
     }
   }
}

Class CSEThread extends Thread {
        @override
        public void run () {
            while (true) {
               System.out.println ("CSE");
              try {
                Thread sleep (2000);
              } Catch (InterruptedException e) {
                  e.printStackTrace();
              }
            }
        }
     }
```

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

        BMSThread bmsThread =new BMSThread();
        bmsThread.start();

        CSEThread cseThread = new CSEThread();
        cseThread.start();

    }
}
```

Output:

```
BMS college of Engineering
CSE
CSE
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
```

[illegible]

Lab-10 :-

```
Class Q
{
  int n;
  boolean valueSet = false;
  Synchronized int get(){
        while (!valueset)
        try {
        System.out.println ("\n Consumer waiting \n");
        wait ();
        } catch (Interrupted Exception e) {
              System.out.println(" Interrupted Exception caught");
        }
    System.out.println("Got: "+n);
    value Set = false;
    System.out.println("\n\n Intimate Producer \n") notify ();
    ~~wait ();~~ return n; }
    [illegible]

Synchronized void put (int n){
  while (valueset)
  try {
    System.out.println("\n Producer waiting \n");
  wait ();
  } Catch (Interrupted Exception e) {
                System.out.println(" Intrupted Exception caught")
  }
```

```
this. n = n;
value Set=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"+x);
            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.");
    }
}
```

→output :-

```
put: 0
interrupt consumer
Producer waiting
Got : 0
Interrupt Producer
Put : 1
Interrupt Consumer
Producer waiting.
Consumed: 0
Got : 1
Interrupt producer
consumed : 1
Put : 2
```

Intimat . produc

consumed : 2 ,

[illegible] 06.02.24

deadlock:-

```
Class A
{
  synchronized void foo (B b)
   {
     String name = Thread. Current ().get Name ();
     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 ();
   }

   void last()
    {
      System. out. println (" Inside A. last");
    }
 }
Class B
{
 Synchronized void bar (A a)
  {
    String name = Thread. current Thread () .get Name ();
    System. out. println (name + " entered B. bar ");
```

```
try
{
  Thread. sleep (1000);
}
Catch (Exception e)
{
    System. out. println ("B Interrupted");
}
. Catch (Exception e)
 {
    System. out. println (Name + " trying to call A. last()");
    a. last();
}
void last()
  {
    System. out. println ("Inside A. last");
  }
}
Class Deadlock implements Runnable
{
   A a = new A();
   B b = new B();
  Deadlock ()
  {
    Thread. Current Thread (). set Name ("Main Thread");
Thread t = new Thread (this, "Racing Thread");
t. start();
a. foo (b);
System. out. println ("Back In main thread");
}
```

```
public void run()
{
    b.bar(a);
    system.out.println ("Back in other thread");
}
public static void main (String args[])
    {
        new Deadlock;
    }
}
```

Output :=

```
Main Thread entered A.foo
Racing Thread entered B.bar
Racing Thread trying to call B.last ()
Inside A.last
Back in main thread
Back in other thread
```

[illegible]
13-02-24

Lab-1

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

class UserInterface {
    UserInterface() {
        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);
        JText field bjtf = new JText Field(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(button);
        jfrm.add (alab)
        jfrm.add (blab);
        jfrm.add (anslab);
ActionListener CalculateListener = new Action Lis
    Public void action Performed (Action Event evt) {
        try {
            int a = Integer . parse Int (ajtf . get Text ());
            int b = Integer . parse Int (bjtf . get Text ());
            if (b==0) {
                throw new Arithmetic Exception ();
            }
            int ans = a/b;
        alab . set Text =("\n A="+a);
        blab . set Text =("\n B ="+ b);
        anslab . set Text ("\n Ans ="+ans);
        err . set Text ("");
    } Catch (Number Format Exception e) {
            display Error Message ("Enter Only Integers !");
      } catch (Arithmetic Exception e) {
            display Error Message ("B should be non - zero !");
      }
  }
private void display Error Message (String message ) {
   alab. set Text ("");
   blab. set Text(""),
  anslab. set Text ("");
  err .set Text (message);
 }
};
```

```
button . add Action Listener (
  jfram . set Visible (true);
}
public static void main (St
      Swing Utilities . invoke L
             public void run
                  new UserI
          });
   }
}
```

→ Output:-

Enter the d

7

Calulat

```
button.addActionListener(calculatListener);
jframe.setVisible(true);

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

→ Output:-

```
Enter the divider and dividend:

    7          4

Calculate     A=7   B=4   Ans = 1
```

[illegible]
10.02.

LAB-1

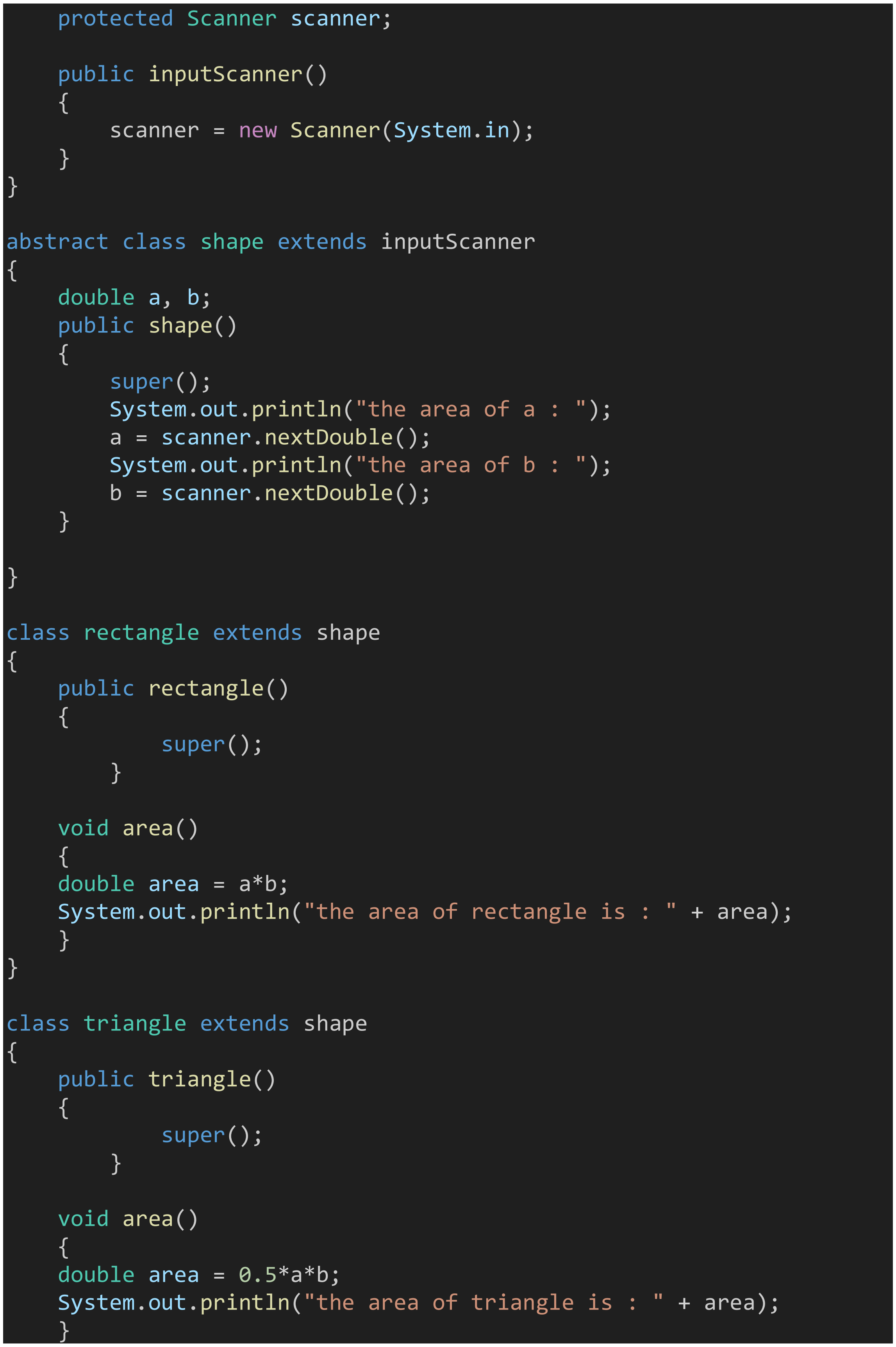
Develop a Java program that prints all real solutions to the
quadratic equation ax 2 +bx+c = 0. Read in a, b,
c and use the quadratic formula. If the discriminate b 2 -4ac is
negative, display a message stating that
there are no real solutions

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

class quadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
        System.out.println("1BM22CS214 REVANTH K");
    }
}
```

```
                            LAB 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.

import java.util.Scanner;

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

class Student
{
    String name;
    String usn;
    double SGPA;
    Scanner s;
    Subject subject[];
    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 your name : ");
        name = s.nextLine();
        System.out.println("enter your usn : ");
        usn = s.nextLine();
    }

    void getMarks()
    {
        int i;
        for(i=0;i<8;i++)
        {
            System.out.println("enter the marks and credits for
course " + (i+1) + ":");
            System.out.println("marks : ");
            int marks = s.nextInt();
            System.out.println("credits : ");
            int credit = s.nextInt();
            subject[i].subjectMarks = marks;
            subject[i].credits = credit;

            if(marks >= 90 && marks<=100)
            {
                subject[i].grade = "O";
            }
            else if(marks>=80 && marks<90)
            {
                subject[i].grade = "A+";
            }
            else if(marks>=70 && marks<80)
            {
                subject[i].grade = "A";
            }
            else if(marks>=60 && marks<70)
            {
                subject[i].grade = "B+";
            }
            else if(marks>=50 && marks<60)
            {
                subject[i].grade = "B";
            }
            else if(marks>=40 && marks<50)
            {
                subject[i].grade = "C";
            }
            else if(marks>=0 && marks<40)
            {
                subject[i].grade = "F";
            }
        }
    }
    void computeSGPA()
    {
        int i;
        double sgpa;
        double totalcredits = 0;
        double totalgradepoints = 0;

        for(i=0;i<8;i++)
        {
            totalcredits += subject[i].credits;
            switch(subject[i].grade)
            {
                case "O" : totalgradepoints +=
10*subject[i].credits;
                break;
                case "A+" : totalgradepoints +=
9*subject[i].credits;
                break;
                case "A" : totalgradepoints += 8*subject[i].credits;
                break;
                case "B+" : totalgradepoints +=
7*subject[i].credits;
                break;
                case "B" : totalgradepoints += 6*subject[i].credits;
                break;
                case "C" : totalgradepoints += 5*subject[i].credits;
                break;
                case "F" : totalgradepoints += 0*subject[i].credits;
                break;
            }
        }
        sgpa = totalgradepoints/totalcredits;
        System.out.println("the sgpa is : "+sgpa);
    }
}
class sgpa
{
    public static void main(String args[])
    {
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
    }
}
```

```
                            LAB 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 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 name,author,price,numPages;
        name="Book name:" +this.name+ "\n";
        author="Author name:" +this.author+ "\n";
        price="Price:" +this.price+ "\n";
        numPages="Number of pages:" +this.numPages+ "\n";
        return name+author+price+numPages;
    }
}

public class Mainbook
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int n;
        int i;
        String name;
        String author;
        int price;
        int numPages;

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

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

        for(i=0;i<n;i++)
        {
            System.out.println("Enter the details of book" + (i+1) +
":");
            System.out.println("Enter the name of the book:");
            name=s.next();
            System.out.println("Enter the author name:");
            author=s.next();
            System.out.println("Enter the price:");
            price=s.nextInt();
            System.out.println("Enter the number of pages:");
            numPages=s.nextInt();

            b[i]=new Books(name,author,price,numPages);
        }

        System.out.println("Book Details:");
        for(i=0;i<n;i++)
        {
            System.out.println(b[i]);
        }
    }
}
```

```
                              LAB 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;

class inputScanner
{
```



```java
    protected Scanner scanner;

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

abstract class shape extends inputScanner
{
    double a, b;
    public shape()
    {
        super();
        System.out.println("the area of a : ");
        a = scanner.nextDouble();
        System.out.println("the area of b : ");
        b = scanner.nextDouble();
    }

}

class rectangle extends shape
{
    public rectangle()
    {
            super();
        }

    void area()
    {
    double area = a*b;
    System.out.println("the area of rectangle is : " + area);
    }
}

class triangle extends shape
{
    public triangle()
    {
            super();
        }

    void area()
    {
    double area = 0.5*a*b;
    System.out.println("the area of triangle is : " + area);
    }
```

```
}

class circle extends shape
{
    public circle()
    {
            super();
        }

    void area()
    {
    double area = 3.14*a*a;
    System.out.println("the area of rectangle is : " + area);
    }
}

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

        rectangle r = new rectangle();
        triangle t = new triangle();
        circle c = new circle();

        r.area();
        t.area();
        c.area();
    }
}
```

```
                              LAB 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.

Create a class Account that stores customer name, account number and
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 minimum 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,cant
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)
        {
```

```java
            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 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.

// Internals.java
package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int marks[] = new int[5];



    public void inputCIEmarks() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Internal Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.nextInt();
        }
    }
}
// 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 scanner = new Scanner(System.in);
        System.out.print("Enter USN: ");
        usn = scanner.next();
        System.out.print("Enter Name: ");
        name = scanner.next();
        System.out.print("Enter Semester: ");
        sem = scanner.nextInt();
    }

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

```java
}



// 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 scanner = new Scanner(System.in);
        System.out.println("Enter SEE Marks for " + name);
        for (int i = 0; i < 5; i++) {
            System.out.print("Subject " + (i + 1) + " marks: ");
            marks[i] = scanner.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("Subject " + (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
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.Scanner;

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

class InputScanner
{
 protected Scanner s;
 public InputScanner()
 {
  s = new Scanner(System.in);
 }
}
```

```
class Father extends InputScanner
{
 protected int fatherAge;
 public Father() throws WrongAge
 {
  System.out.println("Enter Father's Age:");
  fatherAge=s.nextInt();

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

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

}


class Son extends Father
{
 private int sonAge;

 public Son() throws WrongAge
 {
  super();
  System.out.println("Enter Son's age:");
  sonAge=s.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");
  }
 }

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

}


public class FatherSonAge
{
 public static void main(String args[])
 {
  try
  {
   Son son=new Son();
   son.display();
  }

  catch (WrongAge e)
  {
   System.out.println("Error: " + e.getMessage());
  }
 }

}
```

LAB 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 BMSThread extends Thread {
    @Override
    public void run() {
                                    while(true) {
            System.out.println("BMS college of engineering");
            try {
                Thread.sleep(10000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

class CSEThread extends Thread {
    @Override
    public void run() {
        while(true) {
            System.out.println("CSE");
            try {
                Thread.sleep(2000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}

public class threadEx {
    public static void main(String[] args) {
        BMSThread bms = new BMSThread();
        bms.start();
        CSEThread cse = new CSEThread();
        cse.start();
    }
}
```

```
                               LAB 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 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.

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

class UserInterface {
    UserInterface() {
        // create JFrame container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

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

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

        // calc button
```

```java
        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 message
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
        jfrm.add(alab);
        jfrm.add(blab);
        jfrm.add(anslab);

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

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

            private void displayErrorMessage(String message) {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText(message);
            }
        };

        button.addActionListener(calculateListener);
        // 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 UserInterface();
            }
        });
    }
}
```

```
                            LAB 10
Demonstrate Inter process Communication and deadlock

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 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.");
    }
}
```

```
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();

    }

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