

Lab Program 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.

```
import java.util.Scanner;

class Quadratic_eq
{
    public static void main(String args[])
    {
        double a,b,c,d,r1,r2;

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

        Scanner s = new Scanner(System.in);

        a=s.nextDouble();

        b=s.nextDouble();

        c=s.nextDouble();

        d=(b*b)-4*a*c;

        if(a==0)

            System.out.println("invalid input");

        else if(d>0)
        {
```

```

        r1=(-b + Math.sqrt(d))/(2*a);

        r2=(-b - Math.sqrt(d))/(2*a);

        System.out.println("roots are real and distint , values are:" + r1 + "and" + r2);

    }

    else if (d==0)

    {

        r1 = -b/(2*a);

        System.out.println("roots are equal and value is " + r1);

    }

    else

        System.out.println("roots are not real");

}

}

```

OUTPUT:

```

C:\Users\BMSCE\Desktop\1BM21CS060>java Quadratic_eq
Enter the value of coefficient
1
2
1
roots are equal and value is -1.0

C:\Users\BMSCE\Desktop\1BM21CS060>java Quadratic_eq
Enter the value of coefficient
2
1
2
roots are not real

C:\Users\BMSCE\Desktop\1BM21CS060>java Quadratic_eq
Enter the value of coefficient
3
5
2
roots are real and distint , values are:-0.6666666666666666and-1.0

```

WHEN A=0

```
C:\Users\BMSCE\Desktop\IBM21CS060>java Quadratic_eq
Enter the value of coefficient
0
0
0
invalid input
```

Quadratic equation

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

```
Class Quadratic_eq {
```

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

```
{
```

```
    double a, b, c, d, r1, r2 ;
```

```
    System.out.println("Enter the value of coefficient");
```

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

```
    d = s.nextDouble();
```

```
    b = s.nextDouble();
```

```
    c = s.nextDouble();
```

```
    d = (b*b) - 4*a*c ;
```

```
    if (d > 0)
```

```
{
```

```
    r1 = (-b + Math.sqrt(d)) / (2*a);
```

```
    r2 = (-b - Math.sqrt(d)) / (2*a);
```

```
    System.out.println("roots are real and distinct r1 + "and" + r2);
```

```
}
```

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

```
{
```

```
    r1 = -b / (2*a);
```

```
    System.out.println("roots are equal and value is" + r1);
```

```
}
```

else

```
System.out.println("roots are not real");
```

```
}
```

```
}
```

Output:

Enter the value of the coefficient

2

2

2

roots are not real

Enter the value of coefficient

3

5

2

roots are real and distinct, values are: -0.666 and -1.0

Enter the value of coefficient.

1

9

1

roots are equal and value is -1.0

Abeeline

18/11/2022

PROGRAM 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 student
{
    student(){ }
    String name;
    String usn;
    double result=0;
    int credits[]=new int[3];
    int marks[]=new int[3];
    int total=0;
    void accept()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter your Name ");
        name=s.nextLine();
        System.out.println("Enter USN ");
        usn=s.nextLine();
        System.out.println("Enter credits and marks of each subject respectively "); for(int i=0;i<3;i++)
        {
            this.credits[i]=s.nextInt();
            this.marks[i]=s.nextInt();
        }
    }
    void calculate()
    {
        for(int i=0;i<3;i++)
        {
            if(marks[i]>=90 && marks[i]<=100)
                result+=10*credits[i];
            if(marks[i]>=80 && marks[i]<90)
                result+=9*credits[i];
            if(marks[i]>=70 && marks[i]<80)
                result+=8*credits[i];
            if(marks[i]>=60 && marks[i]<70)
                result+=7*credits[i];
            if(marks[i]>=50 && marks[i]<60)
                result+=6*credits[i];
            if(marks[i]>=40 && marks[i]<50)
                result+=5*credits[i];
            else result+=0*credits[i];
        }
        for(int i=0;i<3;i++)
            total+=credits[i];
        result=result/total;
    }
    void display()
    {
```

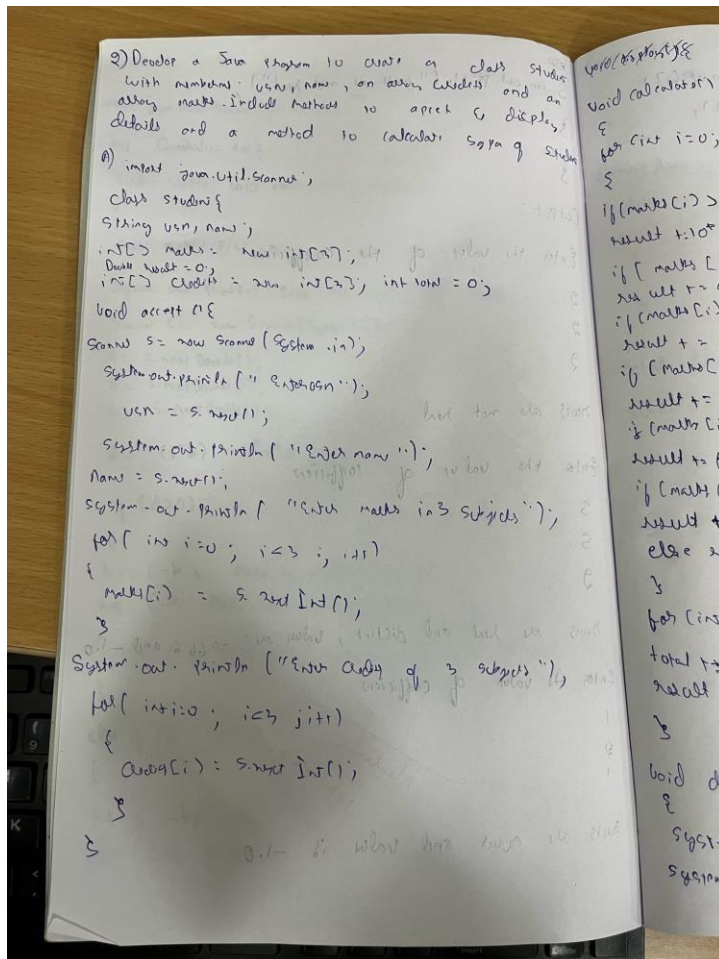


```

System.out.println("Name:"+name+" USN:"+usn);
System.out.println("credits Marks");
for(int i=0;i<3;i++)
System.out.println(credits[i]+" "+marks[i]);
System.out.println("Total credits="+total);
System.out.println("SGPA="+result);
}

```

QUESTION AND CODE



3

3

5100. Out-lying 1" Cubic nails.


```

5 for(int i=0; i<n; i++)
    System.out.println("Chudith " + marks[i]);
    System.out.println("Total Chudith = " + total);
    System.out.println("Sum = " + sum);
}
}

```

class ~~Student~~ Student {

```

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

```

```

        Student s1 = new Student();
        s1.name();
        s1.rollno();
        s1.display();
    }
}

```

OUTPUT

ENTER USN:

18M21CS060

ENTER NAME

Awoiz

Enter marks in 3 arrays

40

80

100

Enter Chudith
4
3
3

Name: Awoiz

Chudith

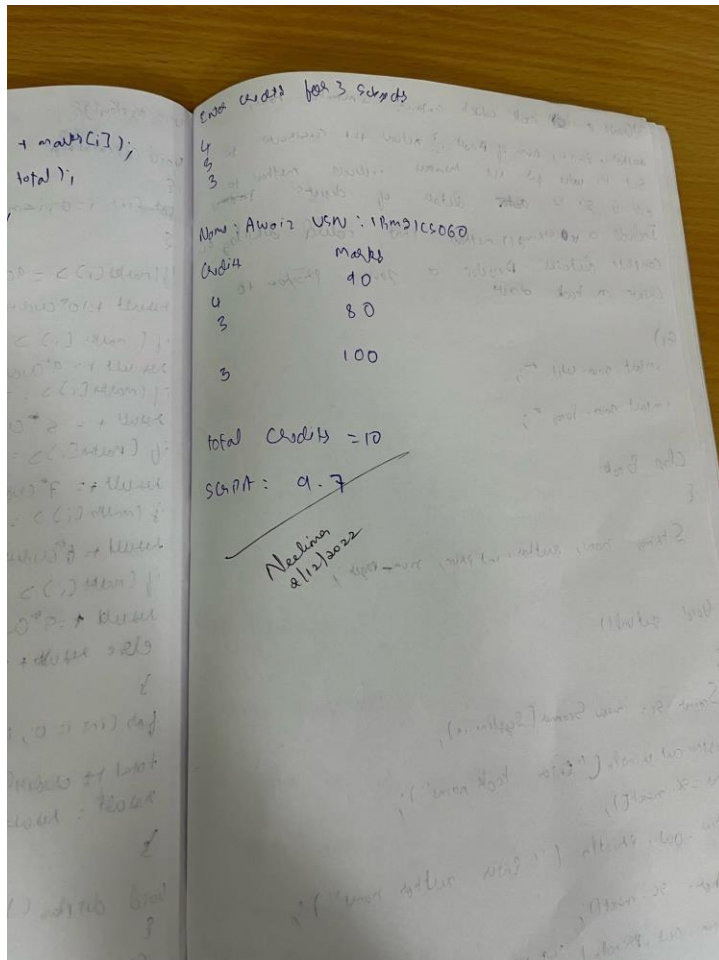
4
3
3

total

sum

total

sum



OUTPUT

```
c:\Users\B1121CS660\Desktop\B1121CS660> java Studentsgpa
Enter username
```

```
B1121CS660
```

```
Enter name
```

```
el/elz
```

```
Enter marks in 3 subjects
98
```

```
88
```

```
100
```

```
Enter credits for 3 subjects
```

```
3
```

```
3
```

```
Name:AOATZUSn:1BN21CS660
```

```
Credits:00nks
```

```
4 90
```

```
3 80
```

```
3 100
```

```
Total credits:10
```

```
SGPA=9.7
```

PROGRAM 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.*;
import java.lang.*;
class Book
{

    String name,author; int price,num_pages;

    void getval()
    {

        Scanner sc=new Scanner(System.in);
        System.out.println("Enter book name");
        name=sc.next();
        System.out.println("Enter author name");
        author=sc.next();
        System.out.println("Enter price ");
        price=sc.nextInt();
        System.out.println("Enter No. of pages");
        num_pages=sc.nextInt();
    }
    public String toString()
    {
        return "name:"+name+" "+"author:"+author+" "+"pric:"+price+" "+"num_pages:"+num_pages+" ";
    }
    void display(Book o)
    {
        System.out.println(o);
    }
}

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

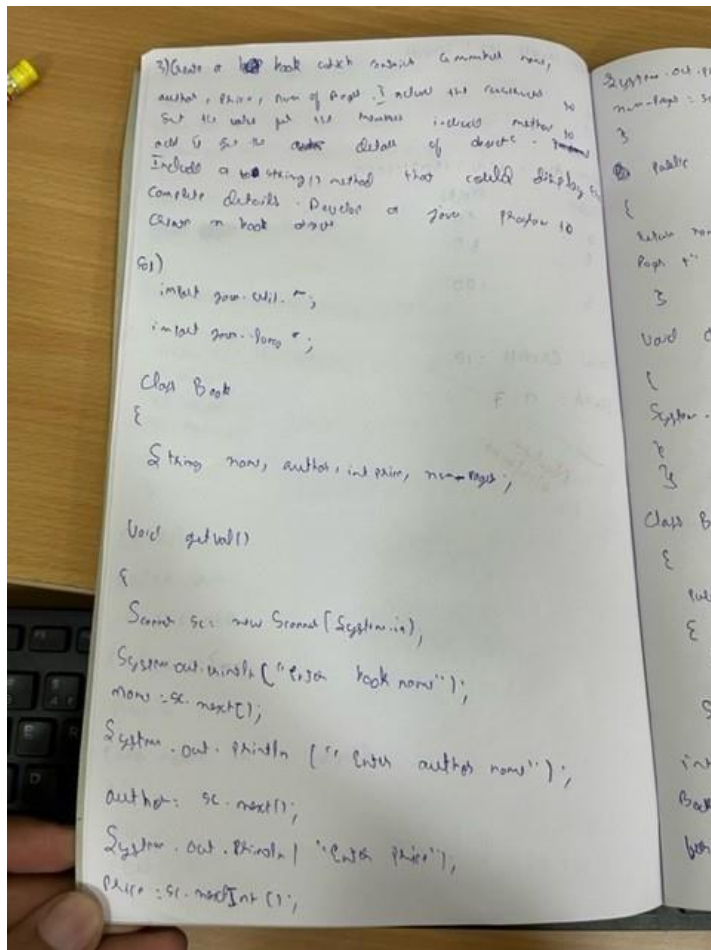
        System.out.println("Enter the no. of book objects");
```

```

int n=in.nextInt();
Book[] ob=new Book[n];
for(int i=0;i<n;i++)
ob[i]=new Book();
for(int i=0;i<n;i++)
{ ob[i].getval();}
for(int i=0;i<n;i++)
{ ob[i].display(ob[i]);}
}
}

```

QUESTION AND CODE



```

    System.out.println("Enter No. of Pages:");
    numPages = sc.nextInt();
}

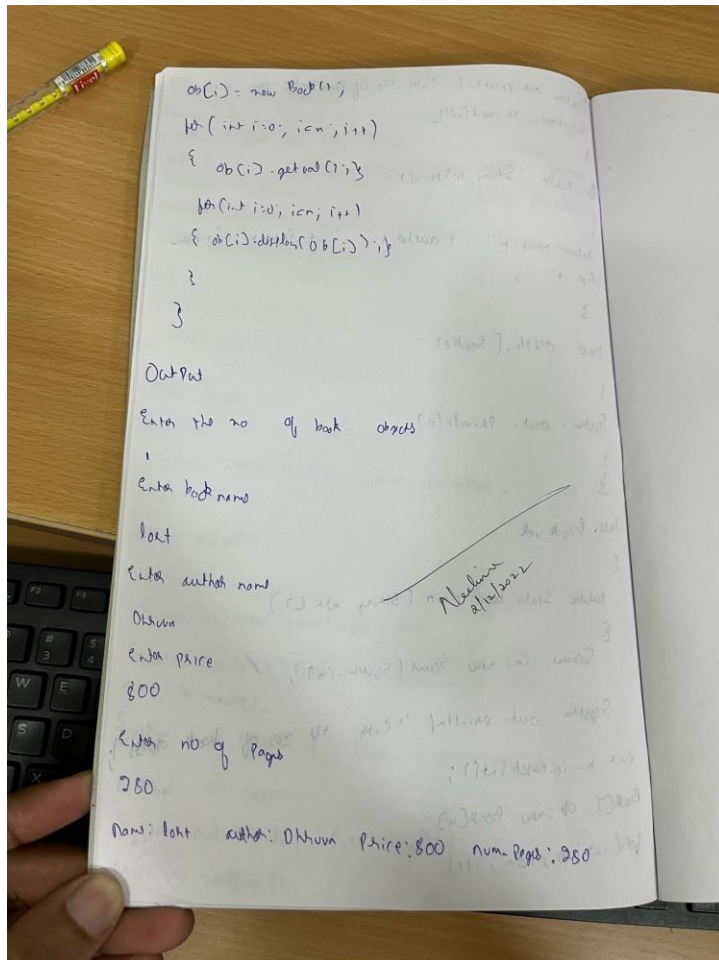
public String toString()
{
    return name + " " + author + " " + title + " " + numPages + " ";
}

void display(Book b)
{
    System.out.println(b);
}

}

Class BookTest
{
    public static void main (String args[])
    {
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the no of book objects:");
        int n = in.nextInt();
        Book[] ob = new Book[n];
        for (int i = 0; i < n; i++)
    
```

OUTPUT

```
C:\Users\8USCE\Desktop\!Bf12iCS060> java OookDET
```

```
Enter the no. of book objects
```

```
2
```

```
Enter book name
```

```
lort
```

```
Enter author name
```

```
awaiz
```

```
Enter price
```

```
800
```

```
Enter No. of pages
```

```
22
```

```
Enter book name
```

```
aaa
```

```
Enter author name
```

```
dhnuva
```

```
Enter price
```

```
300
```

```
Enter No. of pages
```

```
150
```

```
name:lort author:awaiz price:800 num pages:22
```

```
name:aaa author:dhnuva price:300 num pages:150
```

QUESTION AND CODE

PROGRAM 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.

CODE:

```
import java.util.Scanner;
abstract class shape
{
    shape(){}
    int h,b;
    abstract void printArea();
}
class rectangle extends shape
{
    Scanner s=new Scanner(System.in);
    void printArea()
    {
        System.out.println("Enter height and width of rectangle");
        h=s.nextInt();
        b=s.nextInt();
        System.out.println("Area of Rectangle is "+b*h);
    }
    rectangle(){}
}

class triangle extends shape
{
    Scanner s=new Scanner(System.in);
    void printArea()
    {
        System.out.println("Enter height and base of triangle");
        h=s.nextInt();
        b=s.nextInt();
        System.out.println("Area of Triangle is "+0.5*b*h);
    }
    triangle(){}
}

class circle extends shape
{
    Scanner s=new Scanner(System.in);
    void printArea()
    {
        System.out.println("Enter radius of Circle");
        h=s.nextInt();
        System.out.println("Area of Circle is "+3.14*h*h);
    }
}
```

```
}  
circle(){}  
}
```

```
class main  
{  
    public static void main(String xx[])  
    {  
rectangle r=new rectangle();  
r.printArea();  
triangle t=new triangle();  
t.printArea();  
circle c=new circle();  
c.printArea();  
    }  
}
```

Q) Programme to find area of shapes. (Circle, Rectangle, Triangle)

import java.util.Scanner;
abstract class Shape

{
 int x, y;
 double area;
 abstract void printArea();
}

class Rectangle extends Shape

{
 void printArea()
 {
 area = 0.5 * x * y;
 System.out.println("Area of triangle is: " + area);
 }
}

class Circle extends Shape

{
 void printArea()
 {
 area = 3.14 * x * x;
 System.out.println("Area of circle is: " + area);
 }
}

class Area

{

```
System.out.println("Ends radius:");  
c.g = s.nextInt();  
c.printArea();
```

```
}
```

```
else if (fig.equals("Circle") || fig.equals("circle") || fig.equals("CIRCLE"))
```

```
{
```

```
System.out.println("Ends radius:");
```

```
c.r = s.nextInt();
```

```
c.printArea();
```

```
}
```

```
else
```

```
{  
System.out.println("WRONG SHAPE ENTERED!!!");
```

```
}
```

```
}
```

```
}
```



```

public static void main(String args[])
{
    Rectangle r = new Rectangle();
    Triangle t = new Triangle();
    Circle c = new Circle();

    for(int i=0; i<4; i++)
    {
        System.out.println("Enter the shape whose area is to be found out:");
        Scanner s = new Scanner(System.in);
        String fig = s.next();

        if((fig.equals("rectangle") || fig.equals("Rectangle") ||
            fig.equals("RECTANGLE")))
        {
            System.out.println("Enter length:");
            r.l = s.nextInt();
            System.out.println("Enter breadth:");
            r.b = s.nextInt();
            r.printArea();
        }
        else if((fig.equals("triangle") || fig.equals("Triangle") ||
            fig.equals("TRIANGLE")))
        {
            System.out.println("Enter height:");
            t.h = s.nextInt();
        }
    }
}

```

OUTPUT

```

Enter the shape whose area is to be found out: rectangle
Enter length:
3
Enter breadth:
4
Area of rectangle is: 12.0
Enter the shape whose area is to be found out: triangle
Enter height:
4
Enter breadth:
5
Area of triangle is: 10.0
Enter the shape whose area is to be found out: circle
Enter radius:
4
Area of circle is: 50.24

```

PROGRAM 5

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 a 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.

CODE:

```
import java.util.Scanner;
import java.lang.*;
class Account
{
    String name,type;
    int accno;
    double balance;
    void setD()
    {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter customer name: ");
        name=s.next();
        System.out.print("Enter type of account: ");
        type=s.next();
        System.out.print("Enter account number: ");
        accno=s.nextInt();
        System.out.print("Enter bank balance: ");
        balance=s.nextDouble();
    }
    void display()
    {
        System.out.println("Customer name is: "+name);
        System.out.println("Customer account type is: "+type);
        System.out.println("Customer account number is: "+accno);
        System.out.println("Current balance is: "+balance);
    }
    void deposit()
    {
        System.out.print("Enter the amount to be deposited: ");
        Scanner s=new Scanner(System.in);
        double amt=s.nextDouble();
```

```

balance+=amt;
}
}
class Sav_acc extends Account
{
double interest;
void complnt()
{
int time_in_yrs;
float intr_rate_inPerc;
int n; Scanner s=new Scanner(System.in);
System.out.println("Enter time in yrs: "); time_in_yrs=s.nextInt();
System.out.println("Enter rate of interest: "); intr_rate_inPerc=s.nextFloat();
System.out.println("Enter the number of times interest is compounded per year: "); n=s.nextInt();
interest=balance*(Math.pow((1+intr_rate_inPerc/n),(n*time_in_yrs)));
balance+=interest;
}
void withdraw()
{
System.out.println("Enter the amount to be withdrawn: "); Scanner s=new Scanner(System.in);
double amt=s.nextDouble();
if(balance>amt)
{balance-=amt;}
else
{System.out.println("Amount to be withdrawn greater than balance!!!");}
}

}
class Cur_acc extends Account
{
double check_amt;

void debit_check()
{
System.out.print("Enter the check amount: ");
Scanner s=new Scanner(System.in);
check_amt = s.nextDouble();
if(check_amt>balance-10000)
{
System.out.println("Rupees 1000 penalty imposed...Is it ok to proceed? Enter y for yes and n for no");
String option=s.next();
if(option.equals("y")) {balance=balance-check_amt-1000;}
else {System.out.println("no check debited");}
}
else
{
System.out.println("Rupees "+check_amt+" debited"); balance-=check_amt;
}
}
void withdraw()
{
System.out.println("Enter the amount to be withdrawn: "); Scanner s=new Scanner(System.in);

```

```

double amt=s.nextDouble();
if(balance>amt)
{balance-=amt;}
else
{System.out.println("Amount to be withdrawn greater than balance!!!");}
}
}
class Bank
{
public static void main(String ss[])
{
Scanner s=new Scanner(System.in);
String op1,op2;
Sav_acc s1=new Sav_acc();
Cur_acc c1=new Cur_acc();
while(true)
{
System.out.print("Enter the choice: \n1a.Set the values for savings acc\n1b. display\n1c. deposit\n1d.
Interest\n1e. Withdraw\n1f. exit\n");
op1=s.next();
switch(op1)
{
case "1a":s1.setD();
break;
case "1b":s1.display();
break;
case "1c":s1.deposit();
break;
case "1d":s1.complnt();
break;
case "1e":s1.withdraw();
break;
case "1f":System.exit(0);
}
}

```

```

System.out.print("Enter the choice: \n2a.Set the values for current account\n2b. display\n2c.
deposit\n2d. transferCheck\n2e. Withdraw\n2f. exit\n");
op2=s.next();
switch(op2)
{
case "2a":c1.setD();
break;
case "2b":c1.display();
break;
case "2c":c1.deposit();
break;
case "2d":c1.(debit_check);
break;
case "2e":c1.withdraw();
break;

```

```
case "2f":System.exit(0);  
}  
}  
}  
}
```

5) Bank Management

Code:

```
import java.util.Scanner;  
import java.lang.*;
```

class Account

```
{  
    String name, type;  
    int accno;  
    double balance;  
    void set();  
}
```

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

```
System.out.println("Enter Customer name:");
```

```
name = s.next();
```

```
System.out.println("Enter type of account:");
```

```
type = s.next();
```

```
System.out.println("Enter account number:");
```

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

```
System.out.println("Enter book balance:");
```

```
balance = s.nextDouble();
```

```
}
```

```
void display()
```

```
{  
    System.out.println("Customer name is: " + name);
```

```
System.out.println("Customer account type is: " + type);
```

```
System.out.println("Customer account number is: " + accno);
```



```
System.out.println("Bank balance is: " + balance);
}
void deposit()
```

```
{
    Scanner s = new Scanner(System.in);
    double amt = s.nextDouble();
    balance += amt;
}
}
```

```
class Savings extends Account
```

```
{
    double interest;
```

```
    void compInt()
```

```
{
    int time_in_yr;
```

```
    float int_rate_in_pct;
```

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

```
    s.o.p("Enter time in yr:"); time_in_yr = s.nextInt();
```

```
    s.o.p("Enter interest rate:"); int_rate_in_pct = s.nextFloat();
```

```
    s.o.p("Enter no of times interest is compounded for  
    years:"); n = s.nextInt();
```

```
    interest = balance * (Math.pow(1 + int_rate_in_pct/n,
```

```
    (time_in_yr)));
```

```
    balance += interest;
```

```
}
```

```

void withdraw()
{
    S.o.p("Enter amt to be withdrawn:");
    Scanner S = new Scanner(System.in);
    double amt = S.nextDouble();
    if (balance > amt) { balance -= amt; }
    else { S.o.p("Insufficient balance"); }
}
}

```

class Cash - acc extends Amount

```

{
    void min_balance()
    {
        if (balance < 1000)
        {
            S.o.p("Deposit 1000 pents");
            balance += 1000;
        }
        else { S.o.p("No penalty"); }
    }
}

```

void withdraw()

```

{
    S.o.p("Enter amt to be withdrawn:");
    double amt = S.nextDouble();
    if (balance > amt) { balance -= amt; }
}

```

```

else { S.o.p("Insufficient balance"); }
}

```

class Bank

```

{
    private String

```

```

    Scanner

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

    String

```

```

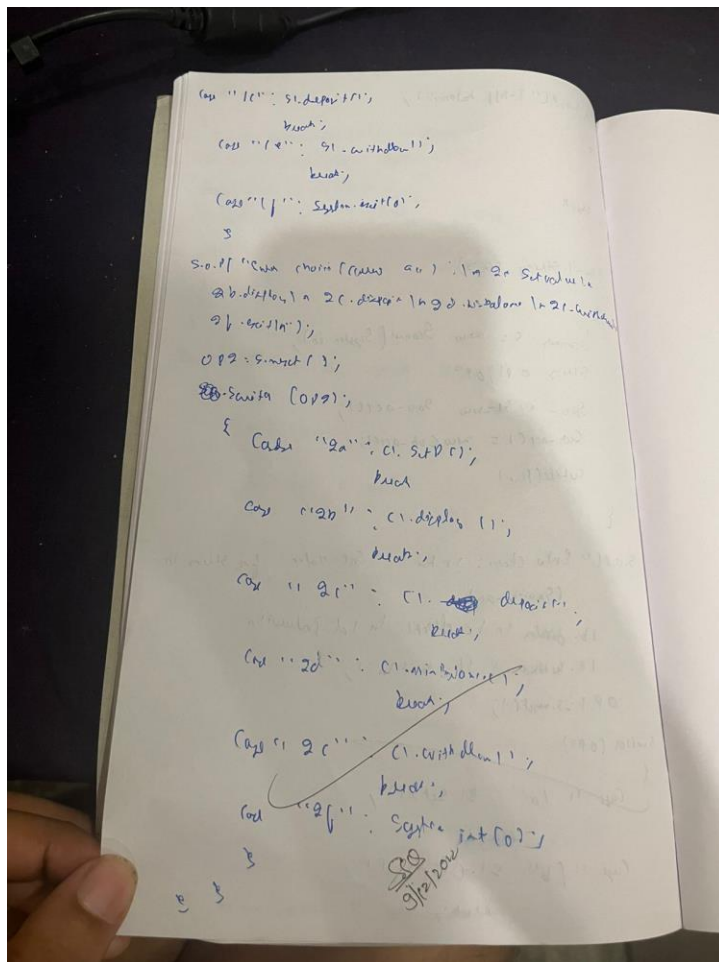
    String

```

```

class { s.o.p ("I-staff balance");
}
3
class Bank
{
    p.gun (String sdc)
    {
        Scanner s = new Scanner (System.in);
        String op, op2;
        Sav = s1 = new Sav-act (1);
        Cur-acc (1) = new Cur-acc (1);
        while (true)
        {
            s.o.p ("Enter chorn: 1 to Set value for stims in
                (Savings-acc)
            1b. deposit in 1c. deposit in 1d. withdrawal
            1e. withdraw 1f. quit\n");
            op = s.next ();
            switch (op)
            {
                case "1a": s1.setD (1);
                    break;
                case "1b": s1.Display (1);
                    break;
            }
        }
    }
}

```



OUTPUT

```

Enter the choice:
1a. Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1a
Enter customer name: dhruva
Enter type of account: saving
Enter account number: 34
Enter bank balance: 34000
Enter the choice:
2a. Set the values for current account
2b. display
2c. deposit
2d. transferCheck
2e. Withdraw
2f. exit
2a
Enter customer name: dhruva
Enter type of account: current
Enter account number: 333
Enter bank balance: 67000
Enter the choice:
1a. Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1b
Customer name is: dhruva
Customer account type is: saving
Customer account number is: 34
Current balance is: 34000.0

```

```
Enter the choice:
2a. Set the values for current account
2b. display
2c. deposit
2d. transferCheck
2e. Withdraw
2f. exit
2b
Customer name is: dhruva
Customer account type is: current
Customer account number is: 333
Current balance is: 67000.0
Enter the choice:
1a. Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1c
Enter the amount to be deposited: 34000
Enter the choice:
2a. Set the values for current account
2b. display
2c. deposit
2d. transferCheck
2e. Withdraw
2f. exit
2c
Enter the amount to be deposited: 30000
Enter the choice:
1a. Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1d
Enter time in yrs:
2
Enter rate of interest:
10
Enter the number of times interest is compounded per year:
3
```

```
Enter the choice:
2a.Set the values for current account
2b. display
2c. deposit
2d. transferCheck
2e. Withdraw
2f. exit
2b
Customer name is: dhruva
Customer account type is: current
Customer account number is: 333
Current balance is: 97000.0
Enter the choice:
1a.Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1b
Customer name is: dhruva
Customer account type is: saving
Customer account number is: 34
Current balance is: 4.5030512967112005E8
Enter the choice:
2a.Set the values for current account
2b. display
2c. deposit
2d. transferCheck
2e. Withdraw
2f. exit
2d
Enter the check amount: 45600
Rupees 45600.0 debited
Enter the choice:
1a.Set the values for savings acc
1b. display
1c. deposit
1d. Interest
1e. Withdraw
1f. exit
1b
Customer name is: dhruva
Customer account type is: saving
Customer account number is: 34
Current balance is: 4.5030512967112005E8
```


CODE

PROGRAM 6

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 WrongAgeException extends Exception{
    public String toString(){
        return ("Negative age can't be accepted");
    }
}

class AgeException extends Exception{
    public String toString(){
        return ("Son can't be older than father");
    }
}

class Father{
    int father_age;
    Father(int x) throws WrongAgeException{
        father_age=x;
        if(father_age<0){
            throw new WrongAgeException();
        }
    }
}

class Son extends Father{
    int son_age;
    Son(int x,int y) throws AgeException, WrongAgeException{
        super(x);
        son_age=y;
        if(son_age<0){
            throw new WrongAgeException();
        }
    }
}
```

```

    }
    if(son_age>=father_age){
        throw new AgeException();
    }
}
}

```

```

class age{
    public static void main(String[] args) {
        try {
            Scanner s=new Scanner(System.in);
            System.out.println("Enter father's and son's ages");
            int x=s.nextInt();
            int y=s.nextInt();
            Son so=new Son(x,y);
            System.out.printf("Father is %d years old and son is %d years old",so.father_age,so.son_age);
        } catch (WrongAgeException wa) {
            System.out.println(wa);
        }
        catch (AgeException a){
            System.out.println(a);
        }
        catch (Exception e){
            System.out.println("Enter valid values");
        }
    }
}
}

```

```

import java.util.Scanner;
class FatherAgeException extends Exception {
    public String toString() {
        return ("Father's age is less than 0");
    }
}

class SonAgeException extends Exception {
    public String toString() {
        return ("Son's age is less than 0");
    }
}

class SonAgeException2 extends Exception {
    public String toString() {
        return ("Son's age is not less than father's age");
    }
}

class Father {
    int age;

    Scanner in = new Scanner(System.in);
    Father() { G.O.P("Enter Father's age: ");
    age = in.nextInt();
}

```

void ex1() throws FatherAgeException

```
{
    if (age < 0)
        throw new FatherAgeException();
}
```

Class Son extends Father

```
{
    int age;
    Son()
    {
        S.o.p("Enter Son's age: ");
        age = in.nextInt();
    }
}
```

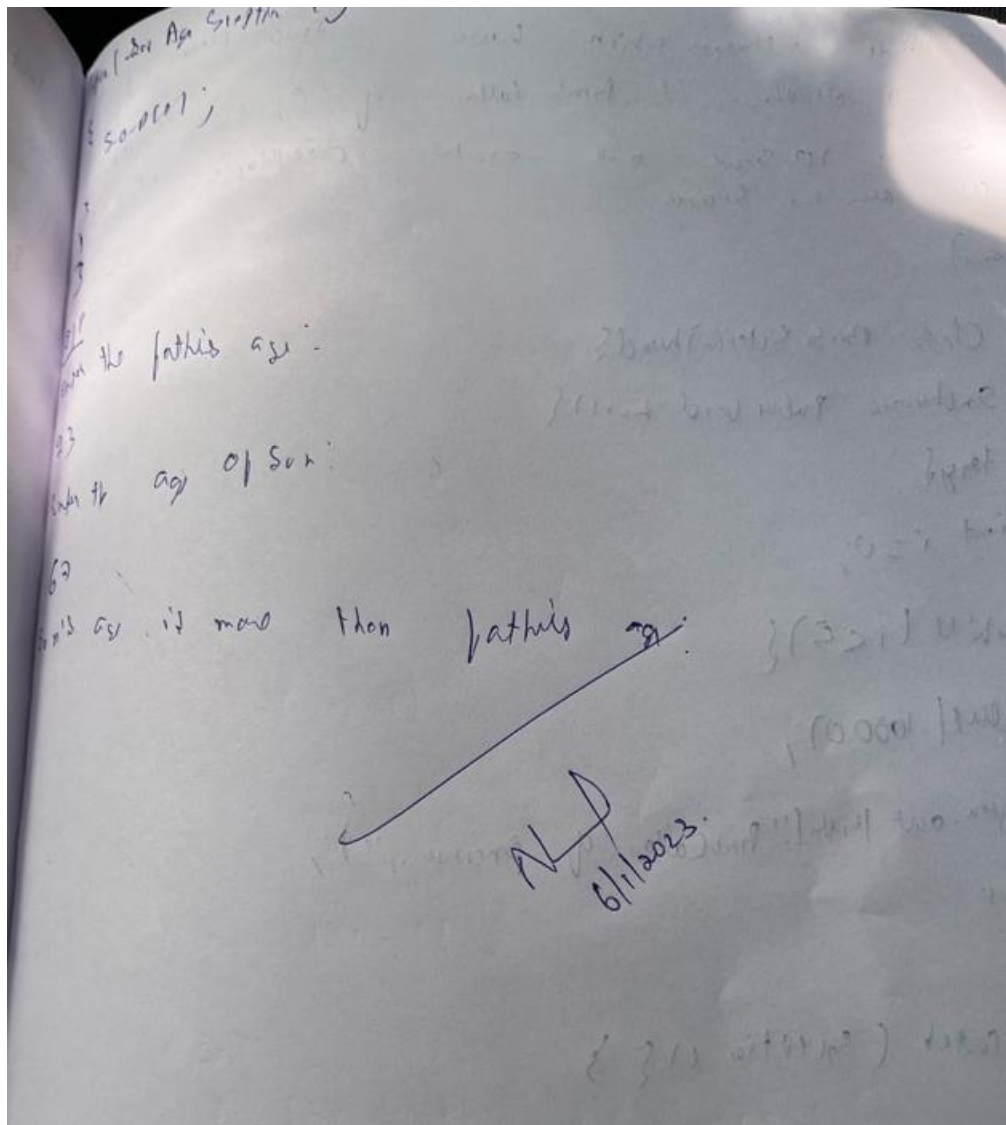
void ex2() throws SonAgeException

```
{
    if (age < 0 || age > 100)
    {
        throw new SonAgeException(age);
    }
}
```

at ex2

pgum (String args[])

```
{
    Son s = new Son();
    try {
        s.ex1();
    }
}
```



OUTPUT

```
C:\Users\BMSCE\Desktop\IBM21CS060>java age
Enter father's and son's ages
22
1
Father is 22 years old and son is 1 years old
C:\Users\BMSCE\Desktop\IBM21CS060>java age
Enter father's and son's ages
22
33
Son can't be older than father
```


Program 7

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.

Code

```
import java.util.Scanner;
class Bms extends Thread {
    synchronized public void run() {
        try {

            int i=0;

            while (i<5) {
                sleep(10000);
                System.out.println("BMS college of engineering ");
                i++;
            }
        } catch (Exception e) {
        }
    }

    class Cse extends Thread {
        synchronized public void run() {
            try {
                int i=0;
                while (i<5) {
                    sleep(2000);
                    System.out.println("Computer science engineering ");
                    i++;
                }
            } catch (Exception e) {
            }
        }
    }

    class Main{
        public static void main(String args[]) {
            Bms t1 = new Bms();
```

```
Cse t2 = new Cse();
```

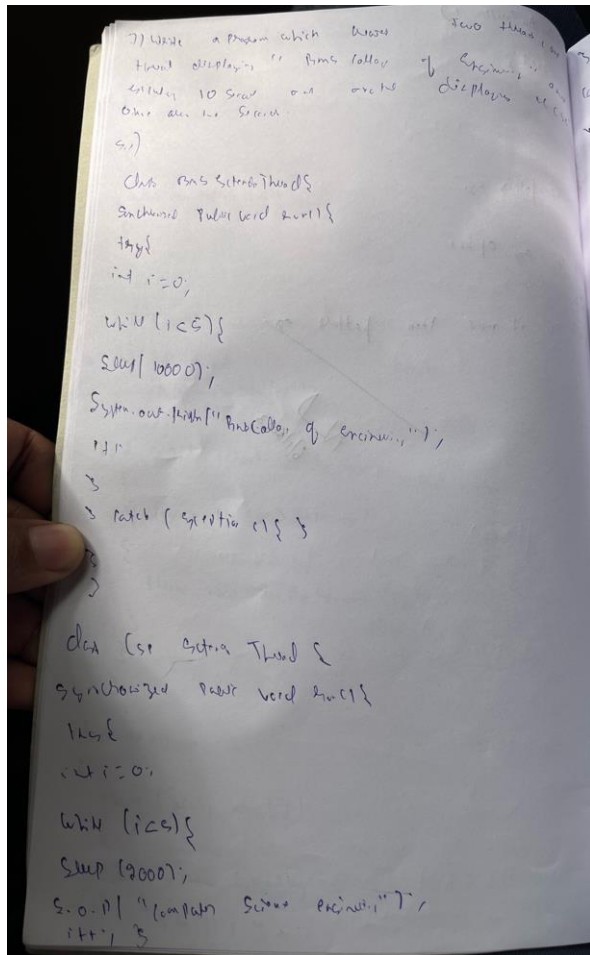
```
t1.start();
```

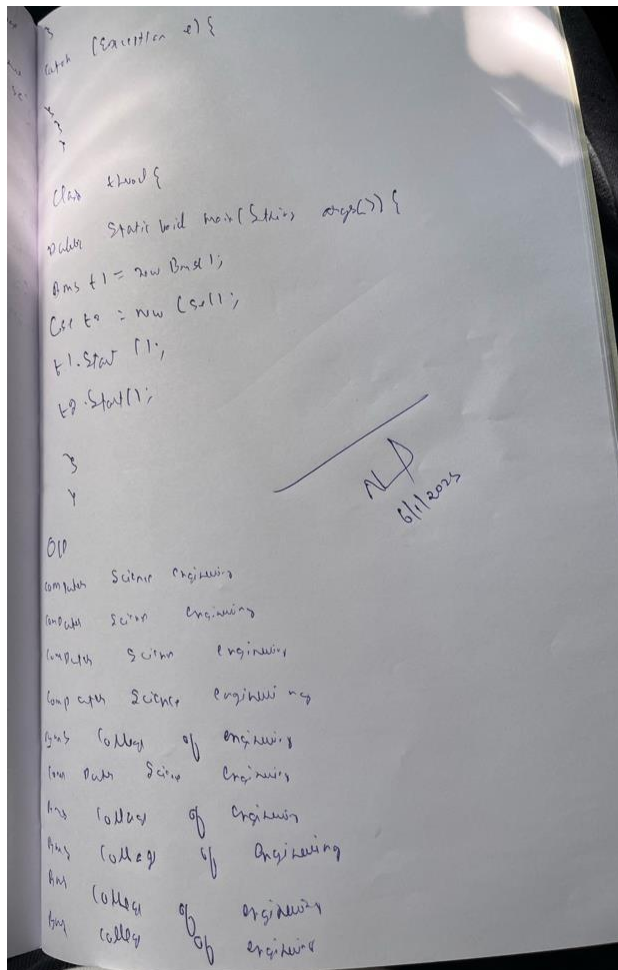
```
t2.start();
```

```
}
```

```
}
```

observation





Output

```

C:\Users\BMSCE\Desktop\1BM21CS060>java thread
Computer science engineering
Computer science engineering
Computer science engineering
Computer science engineering
BMS college of engineering
Computer science engineering
BMS college of engineering
BMS college of engineering
BMS college of engineering
BMS college of engineering
BMS college of engineering

```

Week 8

Create a package CIE

which has two classes- Student and Internals. The class Personal has members

like usn, name, sem. The class internals 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.

CODE

```
package SEE;
import CIE.*;
import java.util.*;
public class externals extends CIE.student{
    Scanner sc=new Scanner(System.in);
    public int seem[]=new int[5];

    public void accept(){
        for(int i=0;i<5;i++)
        {
            System.out.println("Enter SEE marks of subject "+(i+1));
            seem[i]=sc.nextInt();
        }
    }
}

package CIE;
import java.util.*;
public class internals extends CIE.student {
    Scanner sc=new Scanner(System.in);
    public int ciem[]=new int[5];

    public void accept(){
        int i;
        for(i=0;i<5;i++)
        { System.out.println("Enter CIE marks of subject "+(i+1));
            ciem[i]=sc.nextInt();
        }
    }
}
```

```

    }
}
import CIE.*;
import SEE.*;
import java.util.*;
class total{
    public static void main(String args[]) {
        int i,j,n;
        Scanner sc=new Scanner(System.in);
        int total[]=new int[5];
        System.out.println("Enter number of students: ");
        n=sc.nextInt();
        CIE.student s[]=new CIE.student[n];
        CIE.internals ci[]=new CIE.internals[n];
        SEE.externals se[]=new SEE.externals[n];
        for(i=0;i<n;i++)
        {
            System.out.println("\nEnter student "+(i+1)+" details");
            s[i]=new CIE.student();
            s[i].accept();
            ci[i]=new CIE.internals();
            ci[i].accept();
            se[i]=new SEE.externals();
            se[i].accept();
        }
        for(i=0;i<n;i++)
        {
            System.out.println("\nDetails of student "+(i+1));
            s[i].display();
            for(j=0;j<5;j++)
            {
                total[j]=ci[i].ciem[j]+se[i].seem[j];
                System.out.println("Total marks in subject "+(j+1)+" : "+total[j]);
            }
            System.out.println();
        }
    }
}
package CIE;
import java.util.*;
public class student{
    Scanner sc=new Scanner(System.in);
    public String usn,name;
    public int sem;

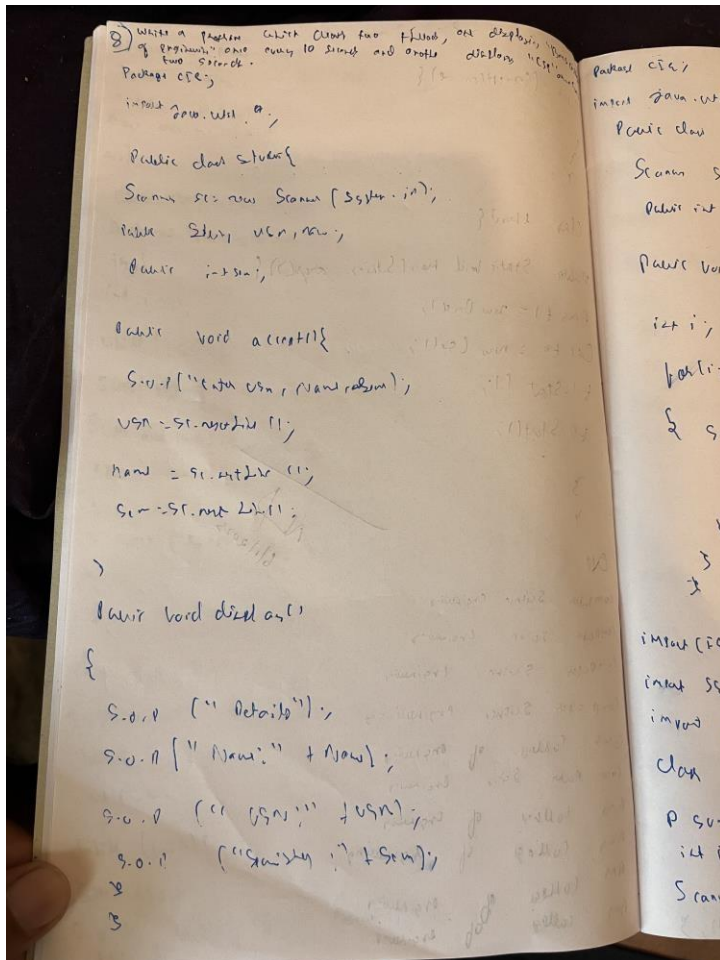
```

```

public void accept(){
    System.out.println("Enter USN, Name and Current semester: ");
    usn=sc.nextLine();
    name=sc.nextLine();
    sem=sc.nextInt();
}

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

```



```

public class ...
    import java.util.*;

    public class ... {
        Scanner sc = new Scanner(System.in);
        public void ... {
            ...
            int i;
            for(i=0; i<5; i++)
            {
                s.o.p("Enter num");
                cin[i] = sc.nextInt();
            }
        }
    }

    import java.util.*;
    import java.io.*;
    class ... {
        public static void main(String args[]) {
            int i;
            Scanner sc = new Scanner(System.in);

```

```

int total[] = new int[5];
s.o.p ("Enter number of Students:");
n = sc.nextInt();
CIL Student s[] = new CIL[n];
CIL student ci[] = new CIL[n];
s.sc.gets(s[i]) = new Sec.expand(ch);
for(i=0; i<n; i++)
{
    s.o.p ("Enter Student "+(i+1)+" details:");
    s[i] = new CIL(s[i]);
    s[i].accept();
    ci[i] = new CIL(s[i].input());
    ci[i].accept();
    s[i] = new Sec.expand();
    s[i].accept();
}
for(i=0; i<n; i++)
{
    s.o.p ("Enter details of Student "+(i+1));
    s[i].display();
}
for(i=0; i<n; i++)
{
    total[i] = ci[i].csum[i] + s[i].ssum[i];
    s.o.p ("Total marks "+(i+1)+" : "+total[i]);
}

```

```

s.o.p (" ");
}
}
}
Output:
Enter details
Vgn = 0
Name =
gen = 3
marks of
24
40
28
11
65
marks
65
56
54
53
66
Octa
Name
Vgn
Se.

```

S.O. 12.1

3

3

3

Output :

ENR details

URN = 060

Name = dhawan

sem = 3

Marks of ODE

24

60

20

11

65

Marks of SEE

65

56

50

53

66

Details of student 1 :

Name: dhawan

URN: 060

sem: 3

Total marks in Sub1: 84

Total marks in Sub2: 96

Total marks in Sub3: 76

Total marks in Sub4: 64

Total marks in Sub5: 33

12/1/2023

OUTPUT

```
Enter number of students:
1

Enter student 1 details
Enter USN, Name and Current semester:
1BM20CS100
NITHIN BS
3
Enter CIE marks of subject 1
47
Enter CIE marks of subject 2
46
Enter CIE marks of subject 3
45
Enter CIE marks of subject 4
43
Enter CIE marks of subject 5
44
Enter SEE marks of subject 1
48
Enter SEE marks of subject 2
47
Enter SEE marks of subject 3
44
Enter SEE marks of subject 4
49
Enter SEE marks of subject 5
50

Details of student 1

Student Details
Name: NITHIN BS
USN: 1BM20CS100
Semester: 3
Total marks in subject 1: 95
Total marks in subject 2: 93
Total marks in subject 3: 89
Total marks in subject 4: 92
Total marks in subject 5: 94
```