```
import Java.util.*;
import Java.math.*;

Public class quadratic
{
    int a,b, c;d;
    float r1; r2;
    void getd()
    {
        Scanner s = new Scanner (System.in);
        a = S.nextInt();
        b = S.nextInt();
        c = S.nextInt();
        d = (b*b) - (4*a*c);
    }

void solve()
{
  If (a==0){

      System.out.Println("Not a Proper Equa");
  }

  Else{
    If (d==0){
    System.out.Println("The roots are real and Equal");
    r1 = (-b)/(2*a);
    System.out.Println("roots are" + r1 + "and" + r2);
    }

  Else if{(d>a){
    System.out.Println("The roots are real & distance");
    r1 = (-b + Math.sqrt(d))/(2*a)
}
```

```
class main{
  PVSM(XX[])
  {
      quadratic q= new quad();
        q.get()
        q.solve()
  }
}
```

Output

```
Enter the co-eff of a,b,c
  12 1
  Root1 = Root2 = 1.0
```

```
import java.util.*;
class students
{
    int m,n,grade,total=0;
    double sum=0,sgpa;
    String USN[] = new String[25];
    float marks[] = new float[25];
    int credits[] = new int[25];
    String name[] = new String[25];
    Scanner s = new Scanner(System.in);

void take()
  {
    System.out.println("Enter no of students");
    n = S.nextInt();
    for(int i=0;i<n;i++){
        S.o.P("Enter name and USN")
        name[i]=s.next();
        USN[i] = S.next();
        S.O.P("Enter marks");
        marks[i]= S.nextfloat();
        S.O.P("Enter credit");
        credits[i] = S.nextInt();
    }
  }

void display()
  {
    S.o.P("details of students")
    for(int i=0;i<n;i++)

    S.O.P(marks[i]+" "+credits[i]);
    calc();
  }
```

```
void calc(){
    if (marks >90)
      grade=10;
   else if (marks<90 && marks>80)
      gradde=9;
   else if (marks<80 && marks>70)
      grade=8;
   else if (marks<70 && marks>60)
      grade =7;

    else
      grade0;
  }
    total = total+credit[j];
    Sum = Sum+(credit[j]*grade);
    Sgpa = Sum/total;
    S.o.p(sgpa);
  }
}

class main
{
   PSVm(xx[])
      student t = new student();
      t.take();
      t.display();
   }
}
```

output

Enter no of students

1

Enter name & USN

Jay

118

Enter no of subs

1

Enter credits → Enter marks

2 90

details of students

1. Jay, 118

SGPA = 10.0

11/11/24

3) import java.util.*;

```
class Book
{
  String name;
  String author;
  Double Price;
  int NO Pages;
  Public Book(String name, String auth, double Price, int num)
  {
      this.name = name;
      this.author = author;
      this.Price = Price;
      this.num = num;
  }

  Scanner S = new Scanner(System.in)
  System.out.Println("Enter the Book name");
  this.name = S.nextLine();
  this.author = S.nextLine();
  this.Price = S.nextint();
  this.numPages = S.Nextint();
}
Public void getdetails
  {
     S.O.P("Book name" + name);
     S.O.P("Author" + author);
     S.O.P("Price" + Price);
     S.O.P("NO of Pages" + numpages);
  }

  Public String toString()
  {
     return name \n + author + \n Price + numPages + ;
  }
```

```
Public class main
{
    Psvm (xxC)
    Scanner s=new Scanner(System.in);
    S.o.p("Enter no of Books");
      Book C)=new Book[n]
      for(int i=0;i<n;i++)
      {
        S.o.p("Enter details");
          Book[i]=St details();
      }
      S.o.p(getdetails());
    }
    S.o.p("Incomplete details");
    for(int i=0; i<n; i++)
      {
        S.o.p("Books" + (i+1) + Books[i].toString());
      }
    }
}
```

output

```
Enter details for Book1
Enter book name : Book one
Enter author name : one
 Price :- 25.5

Details of all Books:

Book 1 :
 Book Name : Book one
 Author : Author one
 Price  : $25.5
 No of Pages : 200

Book 2 :
Book Name : Book Two
Author Name ; Auth Two
Price : $30
No of Pages : 300
```

Lab9 → abstract class usage

```
import java.util.*;
abstract class shape{
    int x,y;
    abstract double Printarea(int a, int b);
}
class rectange extends shape{
    return x*y;
}
class triangle Extends shape{
    return 1/2 * x*y;
}
class circle extends shape{
    return 3.14*x*y;
}
class Main{
    PVSm(xxx){
    int ch=0;
    double area;
    rectangle r1 = new rectangle();
    triangle t1 = new triangle();
    circle c1 = new circle();
    Scanner S = new Scanner(System.in);
    while(ch!=-1){
      switch(ch){
        case1:
          area = r1.Printarea(2,3);
          S.o.P(area);
          break;
        case2:
          area = t1.Printarea(4,5)
          S.o.P(area);
```

```
case 3:
  area = c1.PrintArea(7,1)
    S.O.P (area);
    break;
  }
 }
}
```

output

```
Enter ch:
1
area of rectangle = 56
2
area of traingle = 10
3
area of circle = 21.98
0
Exit
```

21/1/24

Lab-04

```
import java.util.Scanner;
import ui.Student;
import CIE.Internal;
import SEE.External;

class main{
    PSVM (S XX(])
    Scanner sc = new Scanner(System.in)
    String name ;
    int usn ;
    int Sem ;
    S.O.P("Enter no of student");
    int n = sc.nextInt();
    Internals in[] = new Internals[n];
    Externals ex[] = new Externals[n];
    for (int i=0 ; i<n ; i++)
    {
        S.O.P("Enter the name of student" + (i+1) + ");
        name = sc.next();
        S.O.P("Enter USN");
        USN = Sc.nextInt();
        S.O.P("Enter Sem name" + (i+1))
        Sem = sc.nextInt();
        in[i] = new Internals(name, USN, sem);
        ex[i] = new External();
        S.O.P("Enter the internal marks");
        for (int j=0; j <5 ; j++)
            in[i].marks[j] = sc.nextInt();
            System.out.Println("Enter the External marks of
            the Stud is 5 Sub :");
        for (int j=0 ; j<5 ; j++)
            ex[i].emarks[j] = sc.nextInt()
```

3

Student . Java

```
Package CIE;
Package class student {
    Public int USN, sem;
    Public string name;
    Public student() { }
    Public student (String name, int USN, int sem);
    this.name = name;
    this.USN = USN;
    this.sem = sem;
  }
}
```

Internals.Java

```
Package CIE;
Public class Internals extends student {
    Public int marks[] = new int [5]
    Public Internals (String name, int USN, int sem)
    {
        Super(name, USN, sem);
    }
```

External.Java

```
Package SEE;
Public class External Extends CIE.student{
    Public int smarks[]= new int[5];
}
```

output

```
Enter no of students
1
Enter name of student
Jay
Enter USN of student
118
Enter sem of student :
1
Enter the internal marks of student in 5 subjects:
10
20
30
46
50
Enter the external marks of students in 5 subjects:
90
80
70
60
50
```

27/01/24

```
n) import Java.util.Scanner;
class wrongage extends Exception;
{
    Public wrongage(string Error);
    { System.out.Println("Error");}
}
class Father
{
    int age;
    Father(int age) throws wrongage
    {
        If(age<0)
        throw new wrongAge("Father's age cannot be neg");
        this.age = age
    }
}
class Son extends Father
{
    int age;
    Son(int age, int sage) throws wrongage
    {
        Super(Age);
        If(s.age >= age)
        throw new wrongAge("son's age cannot be
                            greater than Father's age")
        this.age=age;
    }
}
```

```
class main
{
    Psvm (String args[])
    {
      Scanner S = new Scanner (System.in)
      try{
        System.out.println ("Enter father's age");
        int f-age = S.nextInt ();
        int s-age = S.nextInt ();
        Son a = new Son (f-age, s-age);
        System.out.println ("Father's age" + f-age);
      }
      S.O.P (Son Sage + S-age);
    catch (wrongAge)
    {  s.o.p ("wrongage entered");
    }
    catch (Exception)
    {  s.o.p ("unexpected error");
    }
}
}
```

output

1) Enter Father's age
45
Enter Son's age
15
Father's age: 45
Son's age: 15

Enter Father's age
-2
Enter Son's age
8
Father age can't be negative

Lab-06

Theardins

```
import Java.util.Scanner;
import Java.lang.*;
class DisplayMessage Extends theard
{

    string mess;
    int delay;
    Public displaymessage(String mess, int delay)
    {
        this.mess = mess;
        this.delay = delay;
    }
    Public void run()
    {
    for(int i=0; i<3; i++){
        S.O.P (mess);
        try{
            Theard.sleep(delay * 1000);
        }
        catch & Interrupted exception e){
            e.PrintStackTrace();
        }
    }
}

Public class ThreardsMain{
        PVSm(string XXC)
    {
        DisplayMessage theard1 = new;
        DisplayMessage ("BMS college of Engineering",10);
```

```
Display message thread2 = new()
Displaymessage("CSE", 2);
threads.start();
threads.start();
    3
3
```

Output

```
Bms College of Engineering
CSE
CSE
CSE
Bms college of Engineering
BMS college of Engineering
```

Bank

a) Accept the deposit from customer and update the Balance
b) Display the Balance
c) compute & deposit interest
d) Permit withdrawal & update the Balance

code

```
class Account {
    Protected String customerName;
    Protected String accountName;
    Protected double balance;
    Public account(String customer, String accountnumber){
        this.customername = customerName;
        this.accountnum = accountnumber;
        this.balance = 0;
    }
    Public void deposit(double amount){
        balance += amount;
        System.out.println("Deposit" + amount + "Successfully");
    }
}

    Public void displaybalance() {
        S.O.P("Balance for this " + accountno + "Balance" + Balance);
    }
}

class SavingAccount extends Account{
    Public SavingAccounts(String customername, accno){
        Super(cust_name, acc_no)
    }
```

```
Public void computeInterest() {
    double interestRate = 0.05;
    double Interest = balance * interestRate;
    balance += Interest;
    S.o.p("Interest of " + Interest);
}

Public void withdraw(amount) {
    If(balance >= amount) {
        balance -= amount
        S.o.p("withdraw of amount" + amount);
}

class CurrentAccount Extends Account {
    Private double minBalance = 1000;
    Public currentAccount(String customerName, String accountNo) {
        Super(customerName, accountNo);
    }

Public void withdraw(double amount) {
    If(balance - amount >= minimumBalance) {
        balance -= amount;
    S.o.p(amount);
    }
}

Private void imposeServiceCharges() {
    double ServiceCharges = 20;
    balance -= ServiceCharges;
    S.o.p(ServiceCharges);
    }
}
```

```
class Bank{
    Psvm (String xx[]){

  SavingAcount Savingaccount= new SavingAccount("Jon" + "SA001");
  Current Acount currentacount= new currentAccount("Jane" + "CA202");

  SavingAcount.depoist(5000);
  Saving Acount.displayBalance();
  Saving Acount. compute interest();
  Saving Acount. withdraw(2000);
  Current Acount.depoist(2000);
  current Acount displaybalance();

  }
}
```

output

Acount ID : SA 101

Balance : $500 Successful

Acc ID : SA102

Balane. 5250P

withdrawal of 200 is Sucessfull

Balance 5050.0

withdrawal of 5000.0 Successfull

Acount number = CA202

Balance 3000.0

19/2/24