Name:	Debjit Ghosal
UID:	2023300065
Experiment No.	2b

AIM:	Constructor Overloading
Program 1	
PROBLEM STATEMENT :	Write a menu driven program to recruit an employee (depending on his performance in various rounds) in some software company using constructor overloading. **Selection Criteria for each post is given below:** 1. **Programmer(Minimum total of 80 marks):-** **Rounds:-** 1. **Course Work** 2. **Aptitude Test** 3. **Technical Test** 4. **Interview** 2. **Team Leader(Minimum total of 85 marks):-** **Rounds:-** 1. **Technical Test** 2. **Interview** 3. **Project Manager(Minimum score 90 marks)** **Rounds:-** 1. **Interview** **Create a class Posting and write 3 constructors to initialize the object and set the parameters and display the employee post according to selection criteria.** **Data members:** - **private int courseWork;**

- **private int AptTest;**
- **private int TechTest;**
- **private int interview;**

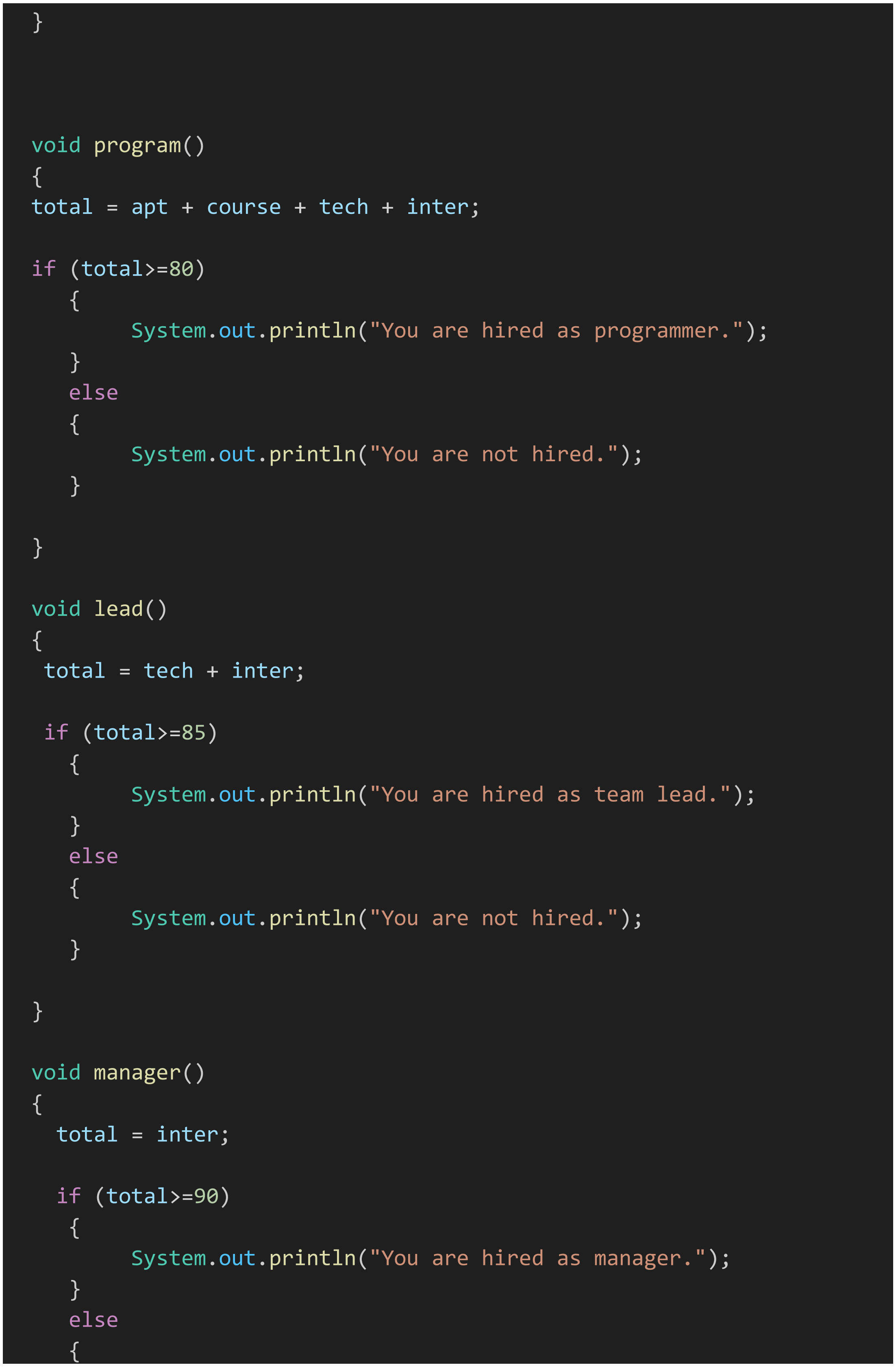
Methods:

- **public Posting(int courseWork, int AptTest, int TechTest,int interview)**
- **public Posting(int TechTest,int interview)**
- **public Posting(int interview)**

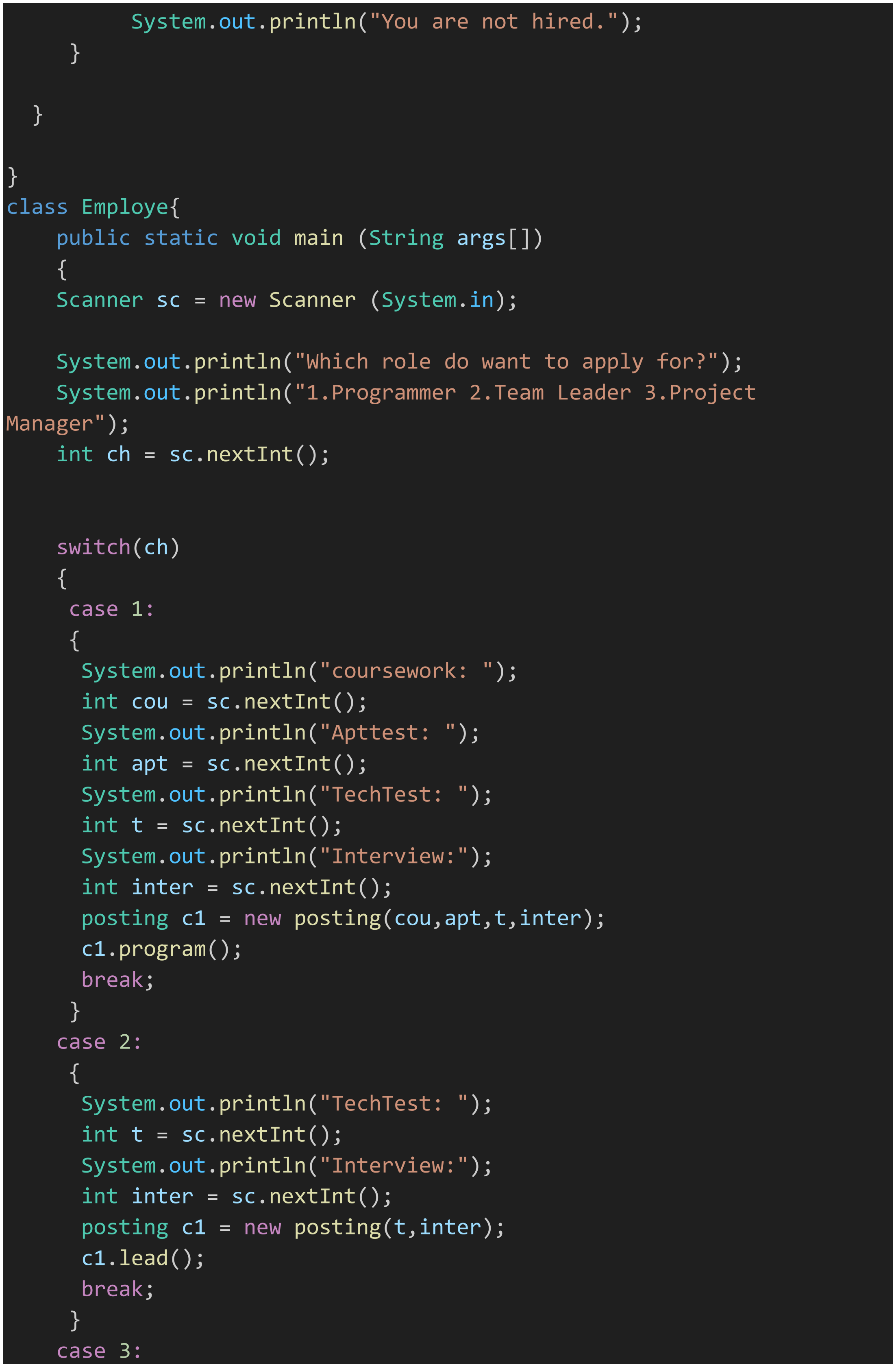
Make use of 'this' keyword. Create arrray of objects , give the list of candidates selected for each post(Programmer, Team Lead and project Manager)

PROGRAM:

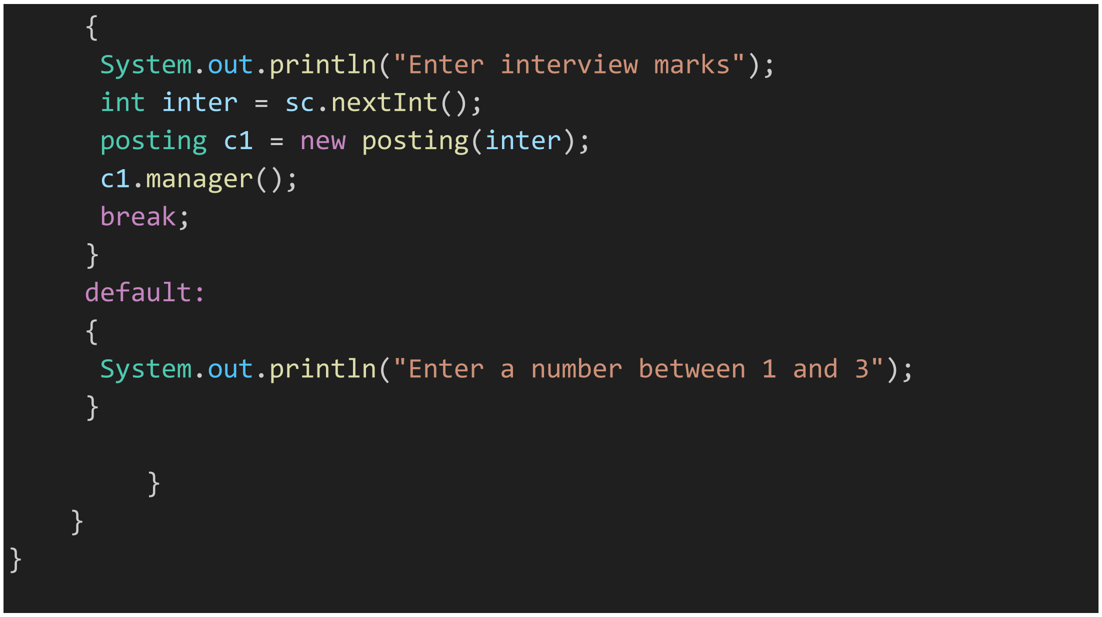
```
import java.util.Scanner;

class posting
{
  int apt;
  int course;
  int tech;
  int inter;
  int total=0;


  posting(int a , int b , int c , int d)
  {
  this.apt = b;
  this.course = a;
  this.tech = c;
  this.inter =d;
  }


  posting (int p , int q)
  {
   this.tech = p;
   this.inter =q;
  }

  posting (int x)
  {
   this.inter =x;
```



```
}



void program()
{
total = apt + course + tech + inter;

if (total>=80)
   {
        System.out.println("You are hired as programmer.");
   }
   else
   {
        System.out.println("You are not hired.");
   }

}

void lead()
{
 total = tech + inter;

 if (total>=85)
   {
        System.out.println("You are hired as team lead.");
   }
   else
   {
        System.out.println("You are not hired.");
   }

}

void manager()
{
  total = inter;

  if (total>=90)
   {
        System.out.println("You are hired as manager.");
   }
   else
   {
```

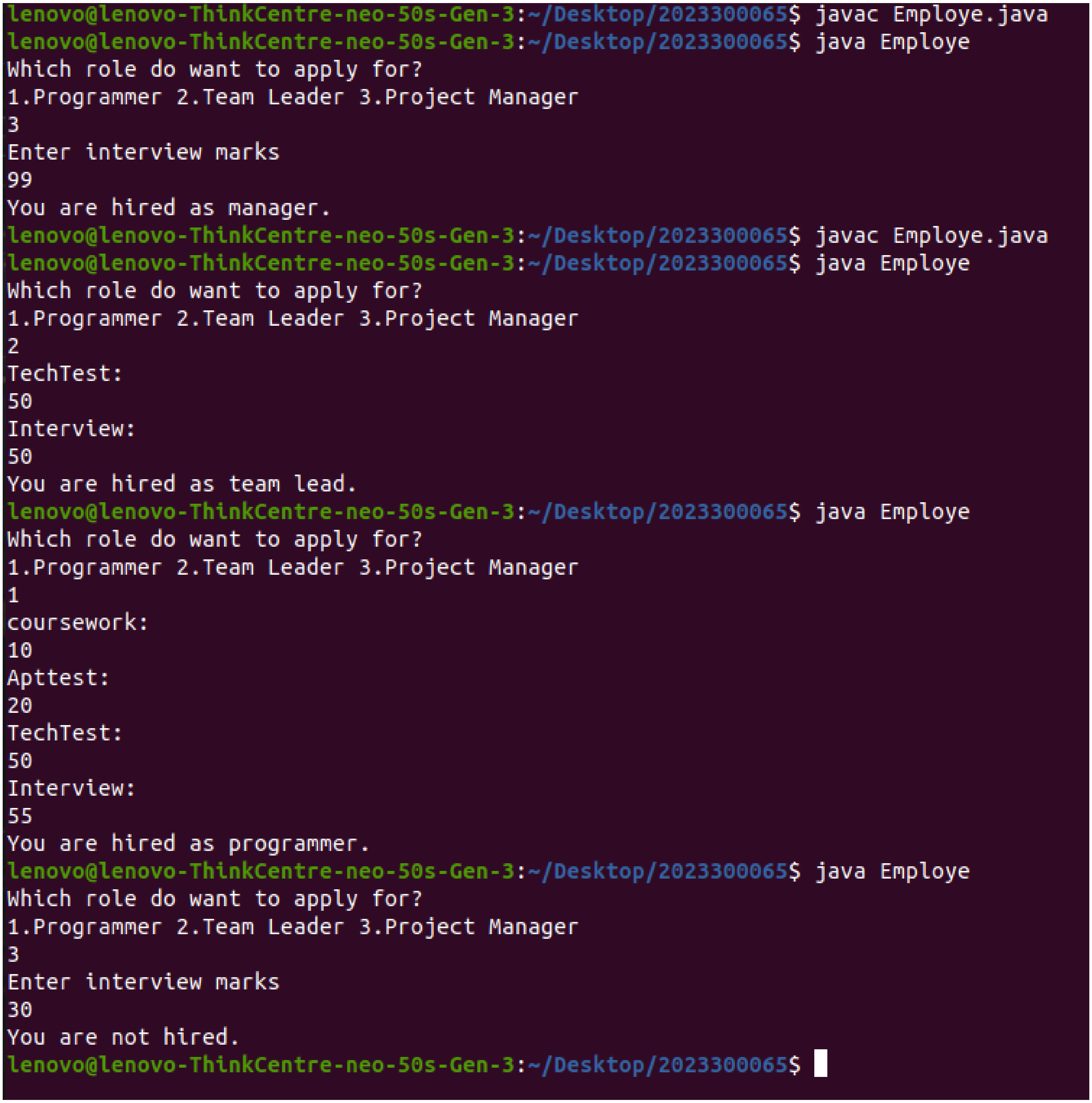


```
          System.out.println("You are not hired.");
     }

  }

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

    System.out.println("Which role do want to apply for?");
    System.out.println("1.Programmer 2.Team Leader 3.Project
Manager");
    int ch = sc.nextInt();


    switch(ch)
    {
     case 1:
     {
      System.out.println("coursework: ");
      int cou = sc.nextInt();
      System.out.println("Apttest: ");
      int apt = sc.nextInt();
      System.out.println("TechTest: ");
      int t = sc.nextInt();
      System.out.println("Interview:");
      int inter = sc.nextInt();
      posting c1 = new posting(cou,apt,t,inter);
      c1.program();
      break;
     }
    case 2:
     {
      System.out.println("TechTest: ");
      int t = sc.nextInt();
      System.out.println("Interview:");
      int inter = sc.nextInt();
      posting c1 = new posting(t,inter);
      c1.lead();
      break;
     }
    case 3:
```



```
     {
      System.out.println("Enter interview marks");
      int inter = sc.nextInt();
      posting c1 = new posting(inter);
      c1.manager();
      break;
     }
     default:
     {
      System.out.println("Enter a number between 1 and 3");
     }

         }
    }
}
```

RESULT:



```
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ javac Employe.java
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java Employe
Which role do want to apply for?
1.Programmer 2.Team Leader 3.Project Manager
3
Enter interview marks
99
You are hired as manager.
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ javac Employe.java
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java Employe
Which role do want to apply for?
1.Programmer 2.Team Leader 3.Project Manager
2
TechTest:
50
Interview:
50
You are hired as team lead.
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java Employe
Which role do want to apply for?
1.Programmer 2.Team Leader 3.Project Manager
1
coursework:
10
Apttest:
20
TechTest:
50
Interview:
55
You are hired as programmer.
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java Employe
Which role do want to apply for?
1.Programmer 2.Team Leader 3.Project Manager
3
Enter interview marks
30
You are not hired.
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ 
```

Program 2	
PROBLEM STATEMENT :	A program to simulate a simple banking system in which the initial balance and rate of interest are read from the keyboard and these values are initialised using the constructor member function. The program consists of the following methods: • To initialise the balance amount and the rate of interest using constructor

member function

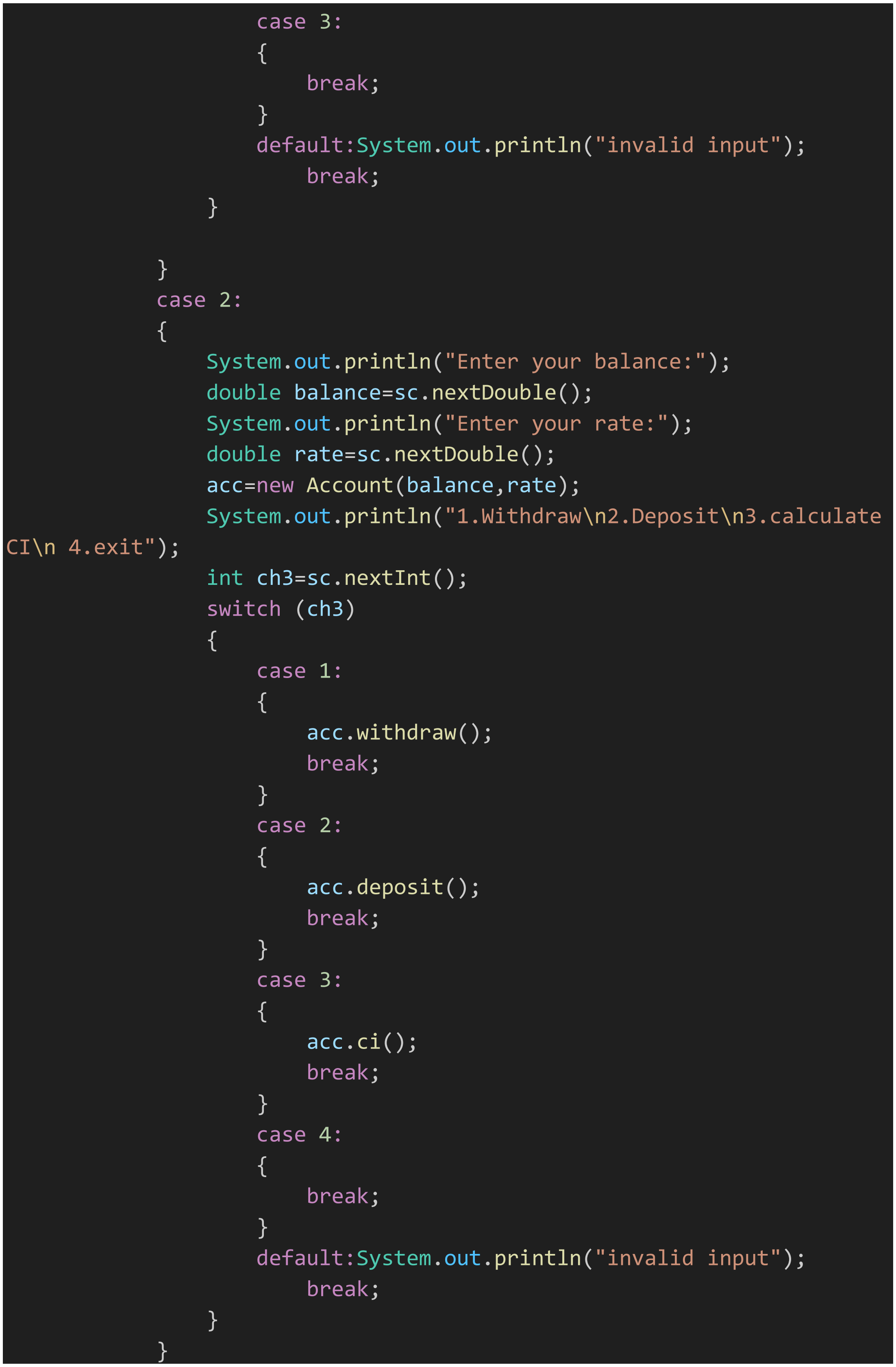
- To make deposit
- To withdraw an amount
- To find compound interest based on the rate of interest
- To know the balance amount
- To display the menu options

Note:

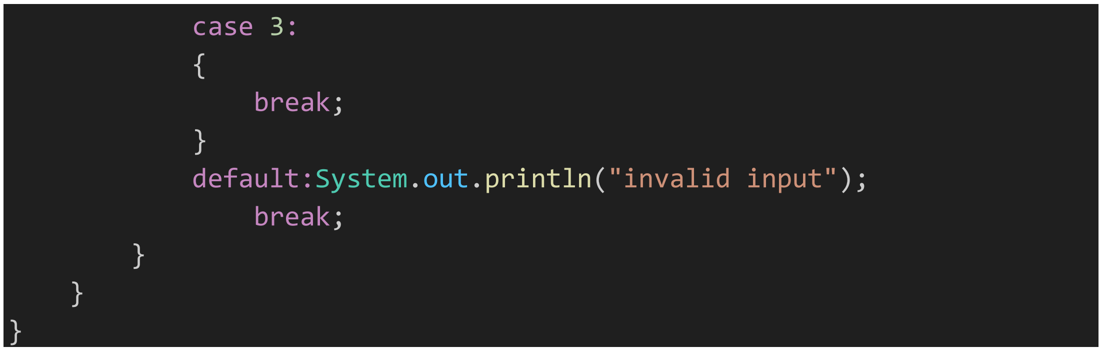
- Balance cannot be less than 0.
- In a Saving account if minBalance is set then for that the balance cannot go less than that amount. If it goes, an error must be shown.
- You can set the values by default for the above variables in Checking Account class

PROGRAM:

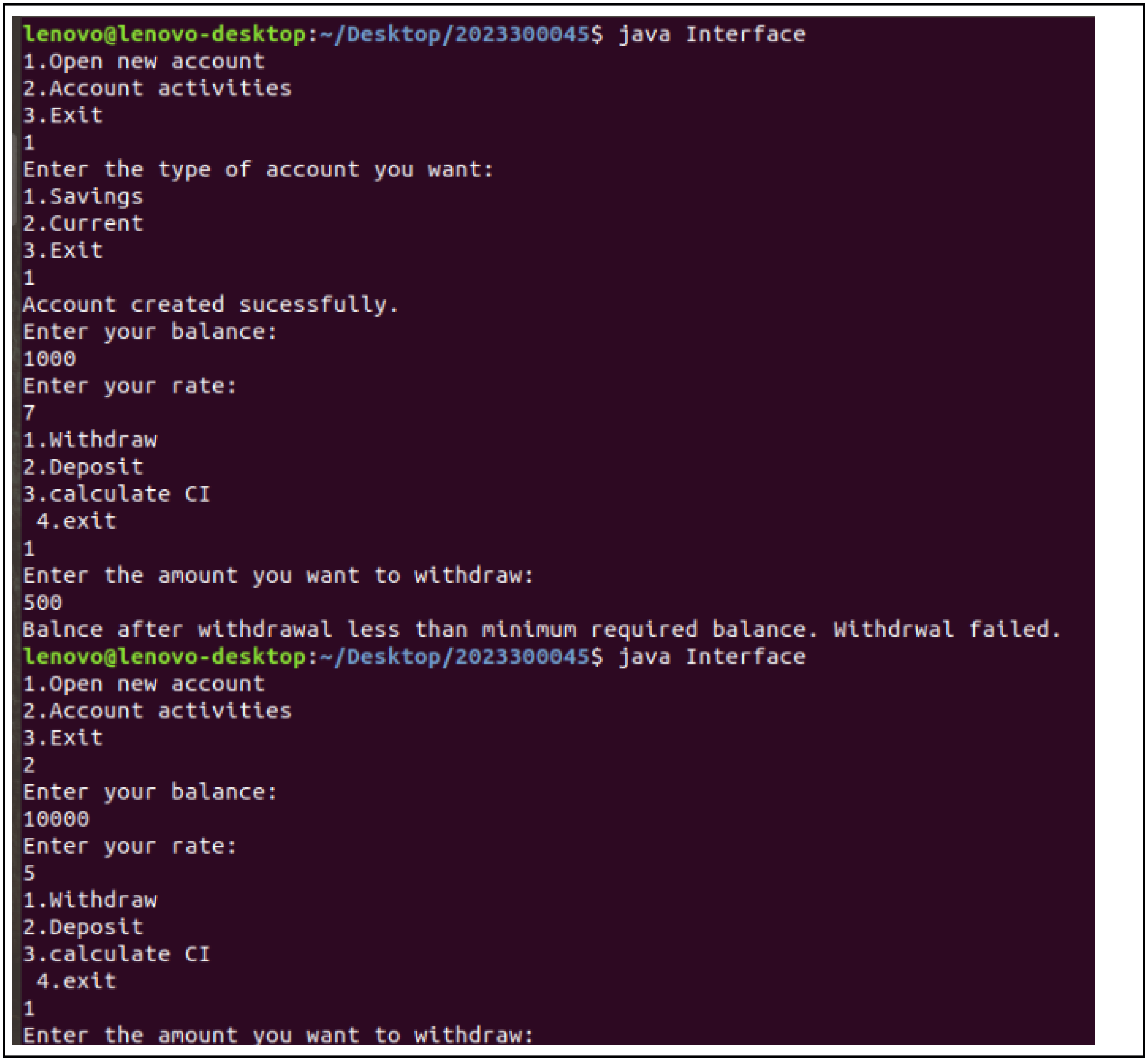
```
import java.util.Scanner;
import java.lang.*;
class Account
{
    double bal;
    double rate;
    double min_bal;
    Account(double bal,double rate)
    {
        this.bal=bal;
        this.min_bal=1000;
        this.rate=rate;
    }
    Account(double min_bal)
    {
        this.bal=1000;
        this.min_bal=min_bal;
        this.rate=7.0;
    }
    Account(double bal,double rate,double min_bal)
    {
        this.bal=bal;
        this.min_bal=min_bal;
        this.rate=rate;
    }
    Account()
    {
        this.bal=1000;
        this.min_bal=1000;
        this.rate=7.0;
    }
    public void withdraw()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount you want to withdraw:");
        double withdraw=sc.nextDouble();
        if(withdraw<bal && (bal-withdraw)>=min_bal)
        {
            bal-=withdraw;
            System.out.println("Cash withdrwal sucessfully done. Your 
balance is now "+bal);
        }
        else if(withdraw>bal)
        {
            System.out.println("Insufficient balance.");
        }
        else if(withdraw<bal && (bal-withdraw)<min_bal)
        {
            System.out.println("Balnce after withdrawal less than 
minimum required balance. Withdrwal failed.");
        }
    }
    public void deposit()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the amount you want to deposit:");
        double deposit=sc.nextDouble();
        bal+=deposit;
        System.out.println("Cash deposit sucessfully done. Your 
balance is now "+bal);
    }
    public void ci()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no. of years you want to 
deposit:");
        int t=sc.nextInt();
        double bal_after = bal*Math.pow((1+(rate/100)),t);
        System.out.println("Your balance after "+t+"years will be 
"+bal_after);
    }
    public void display()
    {
        System.out.println("Your balance is:"+bal);
        System.out.println("ROI is:"+rate);
    }
}


class Interface
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("1.Open new account\n2.Account
activities\n3.Exit");
        int ch1=sc.nextInt();
        Account acc;
        switch (ch1)
        {
            case 1:
            {
                System.out.println("Enter the type of account you
want:");
                System.out.println("1.Savings\n2.Current\n3.Exit");
                int ch2=sc.nextInt();
                switch (ch2)
                {
                    case 1:
                    {
                        acc=new Account();
                        System.out.println("Account created
sucessfully.");
                        break;
                    }
                    case 2:
                    {
                        System.out.println("what rate of intrest do
you want?");
                        double rate=sc.nextDouble();
                        acc=new Account(rate);
                        break;
                    }
```



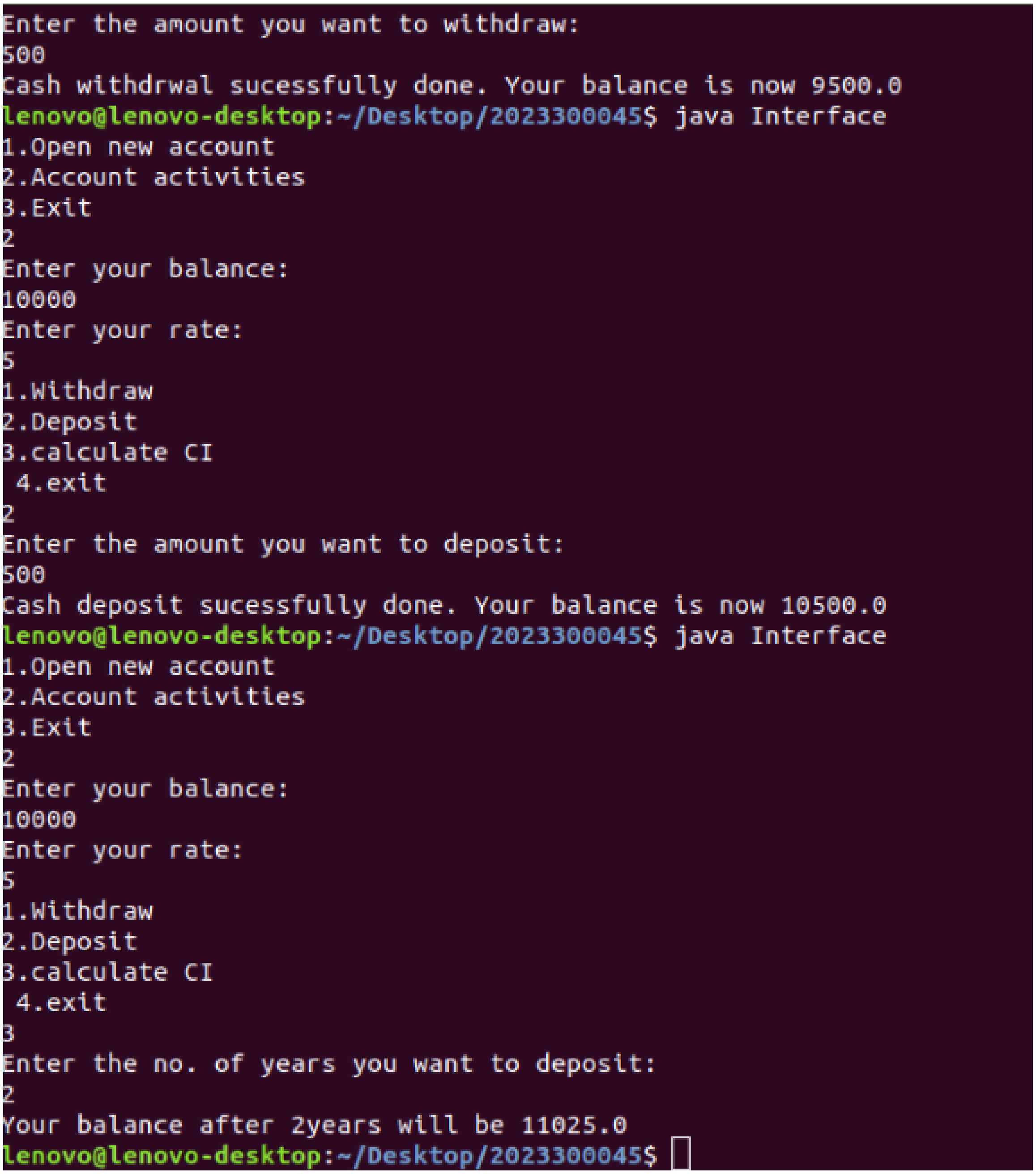
```
                    case 3:
                    {
                        break;
                    }
                    default:System.out.println("invalid input");
                        break;
                }

            }
            case 2:
            {
                System.out.println("Enter your balance:");
                double balance=sc.nextDouble();
                System.out.println("Enter your rate:");
                double rate=sc.nextDouble();
                acc=new Account(balance,rate);
                System.out.println("1.Withdraw\n2.Deposit\n3.calculate
CI\n 4.exit");
                int ch3=sc.nextInt();
                switch (ch3)
                {
                    case 1:
                    {
                        acc.withdraw();
                        break;
                    }
                    case 2:
                    {
                        acc.deposit();
                        break;
                    }
                    case 3:
                    {
                        acc.ci();
                        break;
                    }
                    case 4:
                    {
                        break;
                    }
                    default:System.out.println("invalid input");
                        break;
                }
            }
```

	
RESULT:	

```
            case 3:
            {
                break;
            }
            default:System.out.println("invalid input");
                break;
        }
    }
}
```



```
lenovo@lenovo-desktop:~/Desktop/2023300045$ java Interface
1.Open new account
2.Account activities
3.Exit
1
Enter the type of account you want:
1.Savings
2.Current
3.Exit
1
Account created sucessfully.
Enter your balance:
1000
Enter your rate:
7
1.Withdraw
2.Deposit
3.calculate CI
 4.exit
1
Enter the amount you want to withdraw:
500
Balnce after withdrawal less than minimum required balance. Withdrwal failed.
lenovo@lenovo-desktop:~/Desktop/2023300045$ java Interface
1.Open new account
2.Account activities
3.Exit
2
Enter your balance:
10000
Enter your rate:
5
1.Withdraw
2.Deposit
3.calculate CI
 4.exit
1
Enter the amount you want to withdraw:
```



```
Enter the amount you want to withdraw:
500
Cash withdrwal sucessfully done. Your balance is now 9500.0
lenovo@lenovo-desktop:~/Desktop/2023300045$ java Interface
1.Open new account
2.Account activities
3.Exit
2
Enter your balance:
10000
Enter your rate:
5
1.Withdraw
2.Deposit
3.calculate CI
 4.exit
2
Enter the amount you want to deposit:
500
Cash deposit sucessfully done. Your balance is now 10500.0
lenovo@lenovo-desktop:~/Desktop/2023300045$ java Interface
1.Open new account
2.Account activities
3.Exit
2
Enter your balance:
10000
Enter your rate:
5
1.Withdraw
2.Deposit
3.calculate CI
 4.exit
3
Enter the no. of years you want to deposit:
2
Your balance after 2years will be 11025.0
lenovo@lenovo-desktop:~/Desktop/2023300045$ 
```

CONCLUSION:	I have learnt about the use of constructors, use and importance of ‘this’ keyword and creating array of objects using java.