B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University,
Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of

Computer Science & Engineering (CSE)

Lab Programs Report

Course Title: Object Oriented Java Programming

Course Code: 23CS3PCOOJ

BY

Vaibhav Urs A N (1BM22CS315)

B.M.S. College of Engineering

(Autonomous College Affiliated to Visvesvaraya Technological University,
Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019

Department of
Computer Science & Engineering (CSE)

CERTIFICATE

This is to certify that the report on “**Java Lab Programs**” has been carried out by **Vaibhav Urs** bearing USN **1BM22CS315** as a part of AAT for the course **Object Oriented Java Programming** with course code **23CS3PCOOJ**, Computer Science and Engineering from Visvesvaraya Technological University, Belgaum during the year 2023–24. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report.

Vaibhav Urs A N
1BM22CS315

Shravya AR
Assistant Professor
Department of CSE
BMSCE, Bengaluru-19

Table of contents

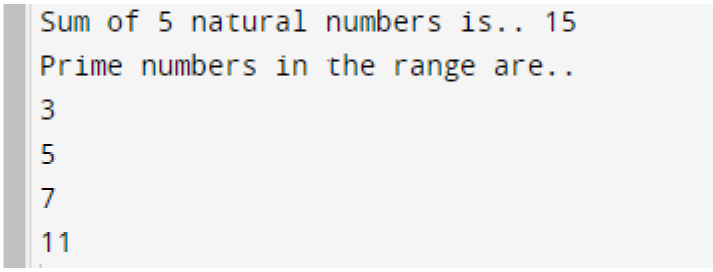
LAB PROGRAM 1

1. Write a program to overload the method print that prints sum of n natural numbers when one variable is passed, and prints the prime numbers in a given range when 2 parameters are passed.

CODE

```
class Overload {

void print(int n) {
int sum = 0;
for(int i=1; i<=n;i++) {
sum = sum+i;
}
System.out.println("Sum of "+n+" natural numbers is "+sum);
}

void print(int m, int n) {

System.out.println("Prime numbers in the range are ");
for(int i=m;i<=n;i++) {
int flag=0;
for(int j=2;j<=i/2;j++) {
if(i%j == 0) {
flag = 1;
break;
}

}
if(flag==0)
System.out.println(i);

}
}
}
class OverloadDemo {

public static void main(String[] args) {
Overload o = new Overload();
o.print(5);
o.print(3,11);
```

OUTPUT



```
Sum of 5 natural numbers is.. 15
Prime numbers in the range are..
3
5
7
11
```

LAB PROGRAM 2

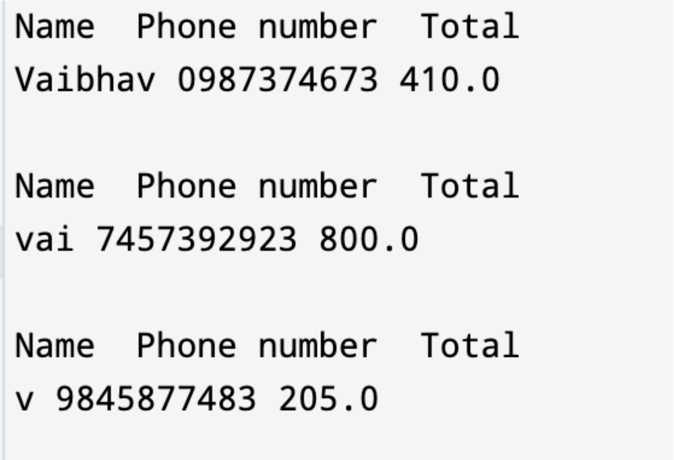
Write a Java program to create a class Grocery that has the variables c_name and c_phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses and quantity of sugar.The method to return the total price. Display the name, ph_no and total bill of 3 customers

CODE

```
class Grocery {
String c_name;
String c_ph;
double total;

Grocery(String c_name, String c_ph){
this.c_name= c_name;
this.c_ph = c_ph;
}

void calc(double q_dal,double q_pulses, double q_sugar) {

total = q_dal*100+q_pulses*80+q_sugar*50;

}

void display()
{
System.out.println("Name "+" "+"Phone number "+" "+"Total");
System.out.println(c_name+" "+c_ph+" "+total);
System.out.println();
}
}

class GDemo {

public static void main(String[] args) {

Grocery g1 = new Grocery("Vaibhav","0987374673");
Grocery g2 = new Grocery("vai","7457392923");
Grocery g3 = new Grocery("v", "9845877483");
g1.calc(2, 2, 1);
g1.display();

g2.calc(3, 5, 2);
```

```
  g2.display();

  g3.calc(1, 1, 0.5);
  g3.display();

  }

  }
```

OUTPUT



```
Name  Phone number  Total
Vaibhav 0987374673 410.0

Name  Phone number  Total
vai 7457392923 800.0

Name  Phone number  Total
v 9845877483 205.0
```

LAB PROGRAM 3

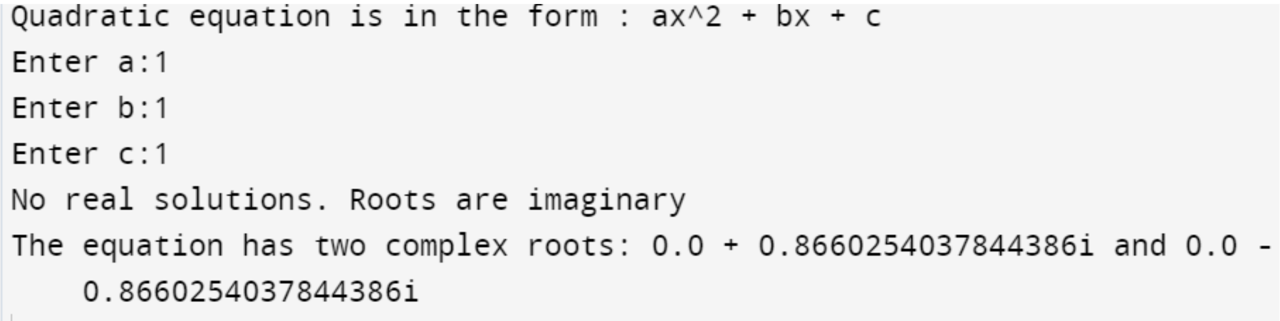
Write a Java program to calculate roots of a quadratic equation. Use appropriate methods to take input, and calculate the roots.

CODE

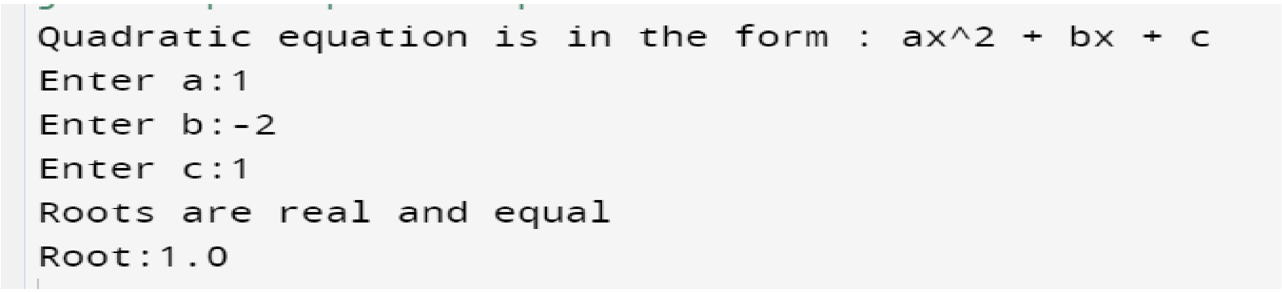
```
import java.util.Scanner;

class Quad {
int a, b, c;
double root1, root2, d;
Scanner s = new Scanner(System.in);

void input()
{

System.out.println("Quadratic equation is in the form : ax^2 + bx + c");

System.out.print("Enter a:");
a = s.nextInt();
System.out.print("Enter b:");
b = s.nextInt();
System.out.print("Enter c:");
c = s.nextInt();
}

void discriminant() {

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

void calculateRoots() {
if(d>0)
{
System.out.println("Roots are real and unequal");
root1 = ( - b + Math.sqrt(d))/(2*a);
root2 = (-b - Math.sqrt(d))/(2*a);
System.out.println("First root is:"+root1);
System.out.println("Second root is:"+root2);
}

else if(d == 0)

{
System.out.println("Roots are real and equal");

root1 = (-b+Math.sqrt(d))/(2*a);
System.out.println("Root:"+root1);
}

else

{
System.out.println("No real solutions. Roots are imaginary");
double real = -b / (2 * a);
double imaginary = Math.sqrt(-d) / (2 * a);
System.out.println("The equation has two complex roots: " + real + " + " + imaginary + "i and "
+ real + " - " + imaginary + "i");
}

}

}
```

```
class Main {

public static void main(String[] args) {
Quad q= new Quad();
q.input();
q.discriminant();
q.calculateRoots();

}
}
```

OUTPUT



```
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:1
Enter c:1
No real solutions. Roots are imaginary
The equation has two complex roots: 0.0 + 0.8660254037844386i and 0.0 -
    0.8660254037844386i
```



```
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:-2
Enter c:1
Roots are real and equal
Root:1.0
```

LAB PROGRAM 4

Create a class Book that contains four members: name, author, price, and 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.

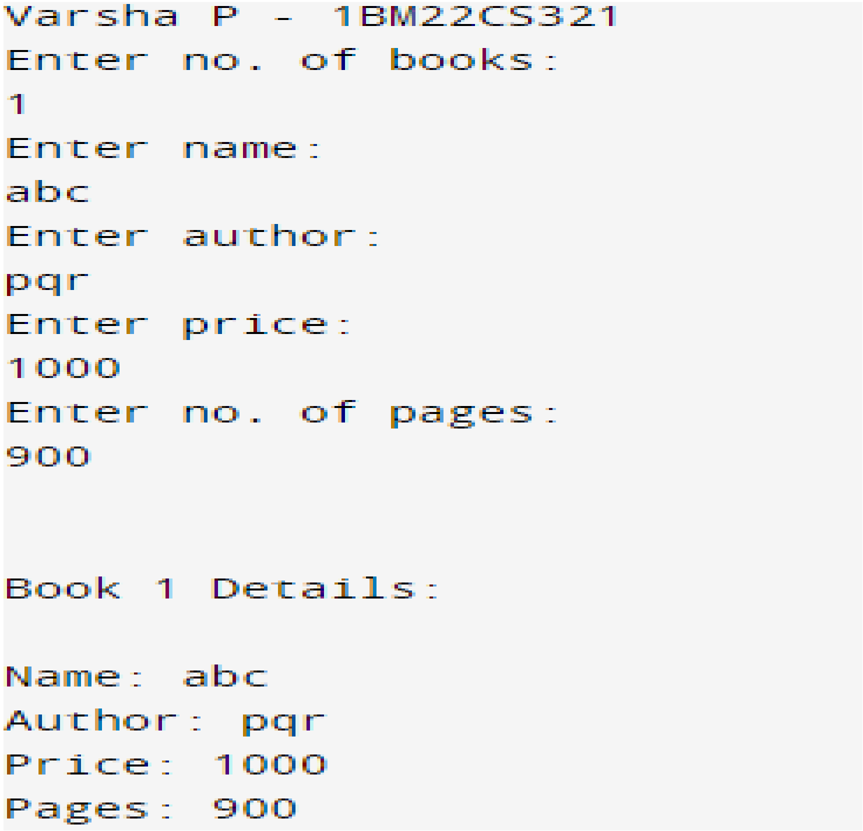
CODE

```
import java.util.Scanner;

class Book{
String name, author;
int price, no_pages;

public Book(String name, String author, int price, int no_pages){
```

```
this.name = name;
this.author = author;
this.price = price;
this.no_pages = no_pages;
}

public String toString(){
System.out.println("Name: " + this.name);
System.out.println("Author: " + this.author);
System.out.println("Price: " + this.price);
System.out.println("Pages: " + this.no_pages);
return this.name + this.author + this.price + this.no_pages;
}
}

class BookMain{
public static void main(String args[]){
System.out.println("Varsha P - 1BM22CS321");
Book books[] = new Book[10];
Scanner sc = new Scanner(System.in);

System.out.println("Enter no. of books: ");
int n = sc.nextInt();
sc.nextLine();
for(int i = 0; i < n; i++){
String name, author;
int price, no_pages;
System.out.println("Enter name: ");
name = sc.next();
System.out.println("Enter author: ");
author = sc.next();
System.out.println("Enter price: ");
price = sc.nextInt();
System.out.println("Enter no. of pages: ");
no_pages = sc.nextInt();
books[i] = new Book(name, author, price, no_pages);
}
System.out.println("\n");
for(int i = 0; i < n; i++){
System.out.println("Book " + (i+1) + " Details:\n");
books[i].toString();
System.out.println("\n");
}
```

}

OUTPUT



```
Varsha P - 1BM22CS321
Enter no. of books:
1
Enter name:
abc
Enter author:
pqr
Enter price:
1000
Enter no. of pages:
900


Book 1 Details:

Name: abc
Author: pqr
Price: 1000
Pages: 900
```

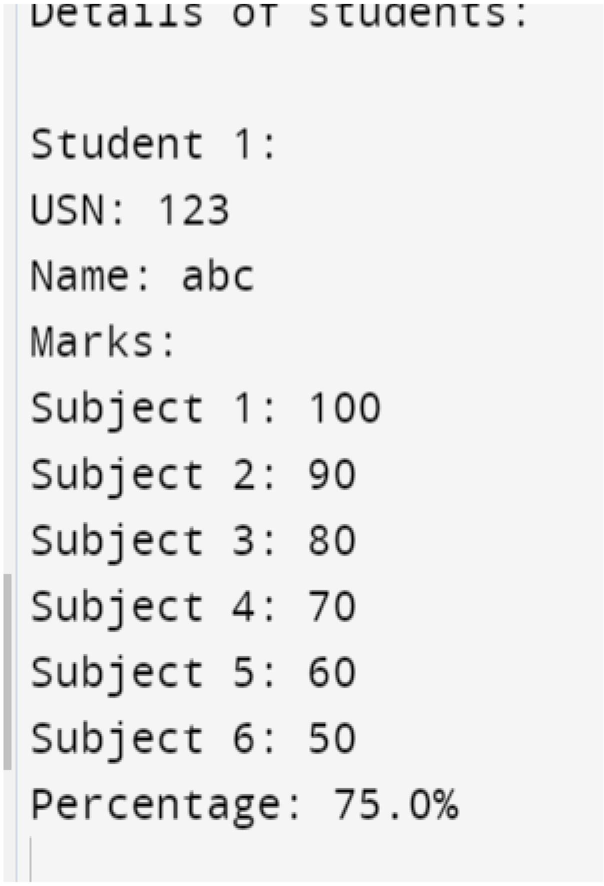
LAB PROGRAM 5

Write a Java program to create a class Student with members USN, name, marks(6 subjects). Include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details. (Array of student object to be created)

CODE

```
import java.util.Scanner;

class Student {
    String USN;
    String name;
    int[] marks = new int[6];

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = scanner.nextLine();
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter marks for 6 subjects:");
        for (int i = 0; i < 6; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public double calculatePercentage() {
        int totalMarks = 0;
        for (int mark : marks) {
            totalMarks += mark;
        }
        return (double) totalMarks / 6;
    }

    public void displayDetails() {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Marks:");
        for (int i = 0; i < 6; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
        System.out.println("Percentage: " + calculatePercentage() + "%");
    }
}

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

        System.out.println("Enter the number of students:");
        int numStudents = scanner.nextInt();
        Student[] students = new Student[numStudents];

        for (int i = 0; i < numStudents; i++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].acceptDetails();
        }

        System.out.println("\nDetails of students:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayDetails();
        }
    }
}
```

OUTPUT



```
Details of students:


Student 1:
USN: 123
Name: abc
Marks:
Subject 1: 100
Subject 2: 90
Subject 3: 80
Subject 4: 70
Subject 5: 60
Subject 6: 50
Percentage: 75.0%
```

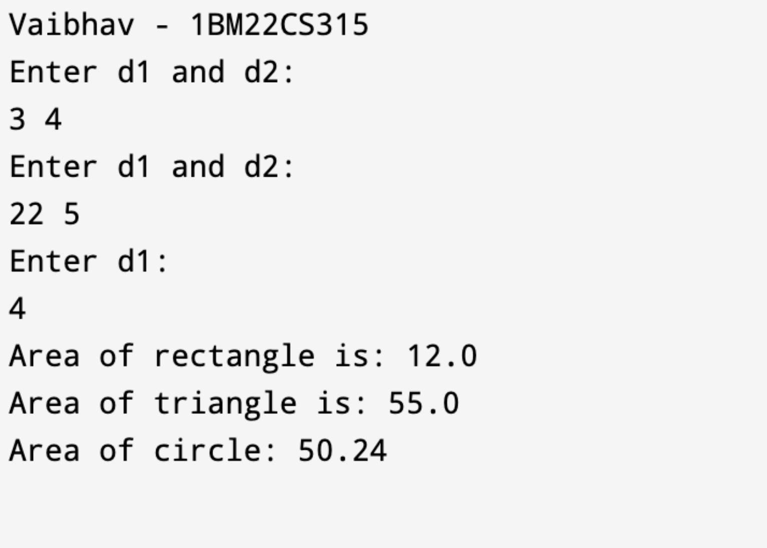
LAB PROGRAM 6

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;

class InputScanner{
int d1, d2;
Scanner sc = new Scanner(System.in);
InputScanner(){
if(this.getClass() == Circle.class){
System.out.println("Enter d1: ");
d1 = sc.nextInt();
}
else{
System.out.println("Enter d1 and d2: ");
d1 = sc.nextInt();
d2 = sc.nextInt();
}
}
}

abstract class Shape extends InputScanner{
abstract void printArea();
}

class Triangle extends Shape{
void printArea(){
System.out.println("Area of triangle is: " + (double)(d1*d2)/2);
}
}

class Rectangle extends Shape{
void printArea(){
System.out.println("Area of rectangle is: " + (double)(d1*d2));
}
}

class Circle extends Shape{
void printArea(){
System.out.println("Area of circle: " + (double)(3.14*d1*d1));
}
}

class AreaMain{
public static void main(String args[]){
System.out.println("Vaibhav - 1BM22CS315");
Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}
```

OUTPUT



```
Vaibhav - 1BM22CS315
Enter d1 and d2:
3 4
Enter d1 and d2:
22 5
Enter d1:
4
Area of rectangle is: 12.0
Area of triangle is: 55.0
Area of circle: 50.24
```

LAB PROGRAM 7

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.

CODE

```
//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 sc=new Scanner(System.in);
System.out.println(&amp;quot;Enter student usn&amp;quot;);
usn=sc.next();
```

```
System.out.println(&amp;quot;Enter student name&amp;quot;);
name=sc.next();
System.out.println(&amp;quot;Enter student semester&amp;quot;);
sem=sc.nextInt();
}
public void displayStudentDetails() {
System.out.println(&amp;quot;student usn:&amp;quot;+ usn);
System.out.println(&amp;quot;student name:&amp;quot;+n ame);
System.out.println(&amp;quot; student sem:&amp;quot;+ sem);
}
}
//Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student {
protected int marks[] = new int[5];
public void inputCIEmarks()
{
Scanner sc=new Scanner(System.in);
for (int i=0;i&amp;lt;5;i++)
{
System.out.println(&amp;quot;Enter 5 subject marks&amp;quot;);
marks[i]=sc.nextInt();
}
}
}
//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 sc = new Scanner(System.in);
for(int i=0;i&amp;lt;5;i++)
{
System.out.print(&amp;quot;Subject &amp;quot;+(i+1)+&amp;quot; marks: &amp;quot;);
marks[i] = sc.nextInt();
}
}
public void calculateFinalMarks() {
for(int i=0;i&amp;lt;5;i++)
finalMarks[i] = marks[i]/2 + super.marks[i];
}
public void displayFinalMarks() {
displayStudentDetails();
```

```
for(int i=0;i&amp;lt;5;i++)

System.out.println(&amp;quot;Subject &amp;quot; + (i+1) + &amp;quot;: &amp;quot; +
finalMarks[i]);
}
}
//Main1.java
import SEE.Externals;
class Main1 {
public static void main(String args[])
{
int numOfStudents = 2;
Externals finalMarks[] = new
Externals[numOfStudents];
for(int i=0;i&amp;lt;numOfStudents;i++)
{
finalMarks[i] = new Externals();
finalMarks[i].inputStudentDetails();
System.out.println(&amp;quot;Enter CIE marks&amp;quot;);
finalMarks[i].inputCIEmarks();
System.out.println(&amp;quot;Enter SEE marks&amp;quot;);
finalMarks[i].inputSEEmarks();
}
System.out.println(&amp;quot;Displaying data:\n&amp;quot;);
for(int i=0;i&amp;lt;numOfStudents;i++)
{

finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
} //end of for loop
} // end of public main
} //end of class main
```

LAB PROGRAM 8

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.

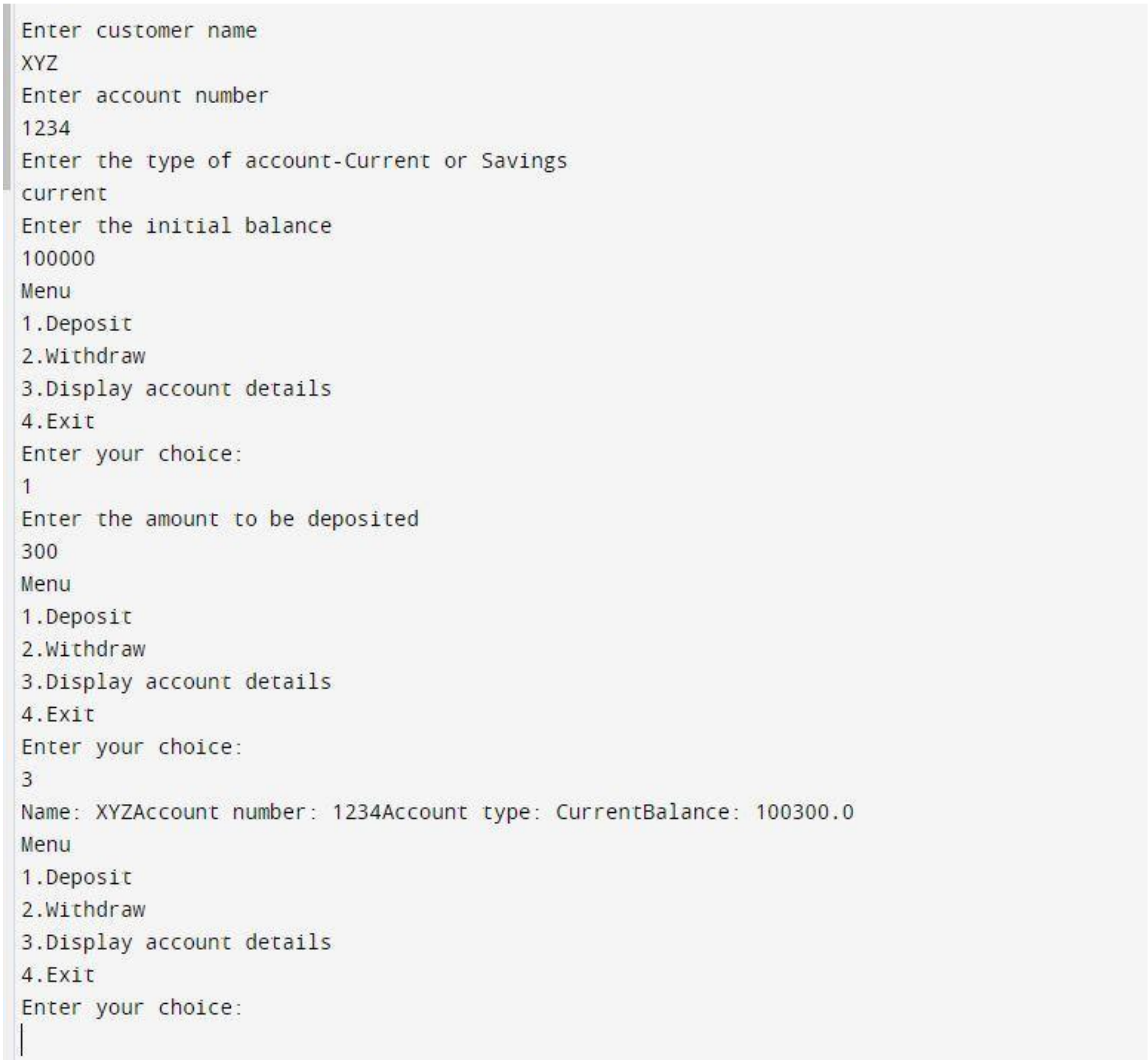
CODE

```
import java.util.*;
class Account{
String name;
int accno;
String acc_type;
double balance;

Account(String name,int accno,String acc_type,double balance)
{
this.name=name;
this.accno=accno;
this.acc_type=acc_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! Cannot withdraw");
}
}
void display()
{
System.out.println("Name:"+" "+name+"Account number:"+" "+accno+"Account type:"+"
"+acc_type+"Balance:"+" "+balance);
}
}

class Sav_acct extends Account
{
static double rate=5.0;
Sav_acct(String name,int accno,double balance)
{
super(name,accno,"Savings",balance);
}

void interest()
{
balance+=balance*(rate)/100;
System.out.println("Balance:"+balance);
}
}

class Curr_account extends Account
{
private double minBal=500;
private double serviceCharges=50;
Curr_account(String name,int accno,double balance)
{
super(name,accno,"Current",balance);
}

void checkMin()
{
if (balance<minBal)
{
System.out.println("Balance is less than minimum, penlaty is imposed"+serviceCharges);
balance-=serviceCharges;
System.out.println("Balance is:"+balance);
}
}
}

class Bank
```

```
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("Enter customer name");
String name=s.next();
System.out.println("Enter account number");
int accno=s.nextInt();
System.out.println("Enter the type of account-Current or Savings");
String acc_type=s.next();
System.out.println("Enter the initial balance");
double balance=s.nextDouble();
Account ob1=new Account(name,accno,acc_type,balance);
Sav_acct sa=new Sav_acct(name,accno,balance);
Curr_account cu=new Curr_account(name,accno,balance);
while (true)
{
if(acc_type.equals("Savings"))
{
System.out.println("Menu");
System.out.println("1.Deposit");
System.out.println("2.Withdraw");
System.out.println("3.Compute Interest for savings account");
System.out.println("4.Display account details");
System.out.println("5.Exit");
System.out.println("Enter your choice:");
int choice=s.nextInt();

switch(choice)
{
case 1:
{
System.out.println("Enter the amount to be deposited");
double amt1=s.nextDouble();
sa.deposit(amt1);
break;
}
case 2:
{
System.out.println("Enter the amount to be withdrawn");
double amt2=s.nextDouble();
sa.withdraw(amt2);
break;
}
```

```
case 3:
{
sa.interest();
break;
}
case 4:
{
sa.display();
break;
}
case 5:
{
break;
}
default:
{
System.out.println("Enter valid input");
break;
}
}

}
else
{
System.out.println("Menu");
System.out.println("1.Deposit");
System.out.println("2.Withdraw");
System.out.println("3.Display account details");
System.out.println("4.Exit");
System.out.println("Enter your choice:");
int choice=s.nextInt();
switch(choice)
{
case 1:
{
System.out.println("Enter the amount to be deposited");
double amt1=s.nextDouble();
cu.deposit(amt1);
break;
}
case 2:
{
System.out.println("Enter the amount to be withdrawn");
double amt2=s.nextDouble();
cu.withdraw(amt2);
break;
}
case 3:
{
cu.display();
cu.checkMin();
break;
}
case 4:
{
break;
}
default:
{
System.out.println("Enter valid input");
break;
}
}

}
}
}
}
```

OUTPUT



```
Enter customer name
XYZ
Enter account number
1234
Enter the type of account-Current or Savings
current
Enter the initial balance
100000
Menu
1.Deposit
2.Withdraw
3.Display account details
4.Exit
Enter your choice:
1
Enter the amount to be deposited
300
Menu
1.Deposit
2.Withdraw
3.Display account details
4.Exit
Enter your choice:
3
Name: XYZAccount number: 1234Account type: CurrentBalance: 100300.0
Menu
1.Deposit
2.Withdraw
3.Display account details
4.Exit
Enter your choice:
```

LAB PROGRAM 8

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.

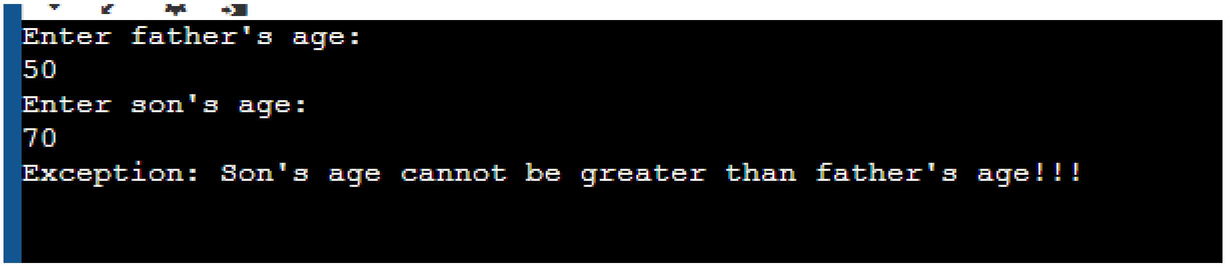
CODE

```
import java.util.Scanner;

class WrongAge extends RuntimeException {
public WrongAge() {
```

```
super("Age cannot be negative");
}

public WrongAge(String message) {
super(message);
}
}

class InputScanner {
protected Scanner scanner;

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

public int nextInt() {
return scanner.nextInt();
}
}

class Father extends InputScanner {
protected int fatherAge;

public Father() {
System.out.println("Enter father's age:");
fatherAge = super.nextInt();

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

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

class Son extends Father {
private int sonAge;

public Son() {
super();
System.out.println("Enter son's age:");
sonAge = super.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 InheritanceException {
public static void main(String[] args) {
try {
Son son = new Son();
son.display();
} catch (WrongAge e) {
System.out.println("Exception: " + e.getMessage());
}
}
}
```

OUTPUT



```
Enter father's age:
50
Enter son's age:
70
Exception: Son's age cannot be greater than father's age!!!
```

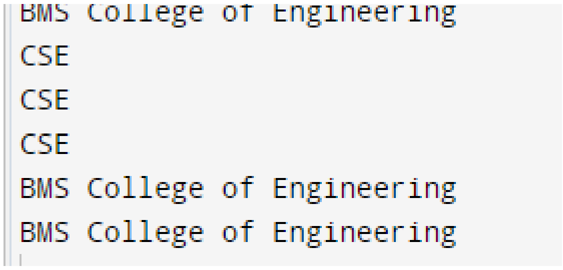
LAB PROGRAM 9

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.io.*;

class B extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("BMS College of Engineering");
Thread.sleep(10000);
}
} catch (InterruptedException e){
System.out.println(e);
}

}
}

class C extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("CSE ");
Thread.sleep(2000);
}
} catch (InterruptedException e){
System.out.println(e);
}

}
}

class ThreadMain{
public static void main(String args[]){
B b = new B();
C c = new C();
b.start();
c.start();
}
}
```

OUTPUT



```
BMS College of Engineering
CSE
CSE
CSE
BMS College of Engineering
BMS College of Engineering
```

THANK YOU