**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**

****

**LAB REPORT**

**on**

**Object Oriented Java Programming**

***Submitted by***

**RYAN THAPA(1BM21CS174)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

**October-2022 to Feb-2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “JAVA LAB PROGRAMS” carried out by **RYAN THAPA(1BM21CS174)** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Object Oriented Java Programming (21CS33PCOOJ)** work prescribed for the said degree.

Name of the Lab-Incharge-             **Dr. Jyothi S Nayak**

**Dr. RAJESHWARI B S**

Designation- Assistant Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

`

WEEK:1

Develop a Java program that prints all real solutions to the quadratic equation ax2+bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

Code:

import java.util.Scanner;

import java.lang.Math;

class QuadraticEquation{

public static void main(String[] args){

int a,b,c;

double d,r1,r2;

Scanner in=new Scanner(System.in);

System.out.println("Enter the values of a,b,c");

a=in.nextInt();

b=in.nextInt();

c=in.nextInt();

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");

System.out.println("The Roots are : "+r1+" "+r2);

}

else if(d==0){

r1=r2=(-b)/(2\*a);

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

System.out.println("The roots are : "+r1);

}

else{

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

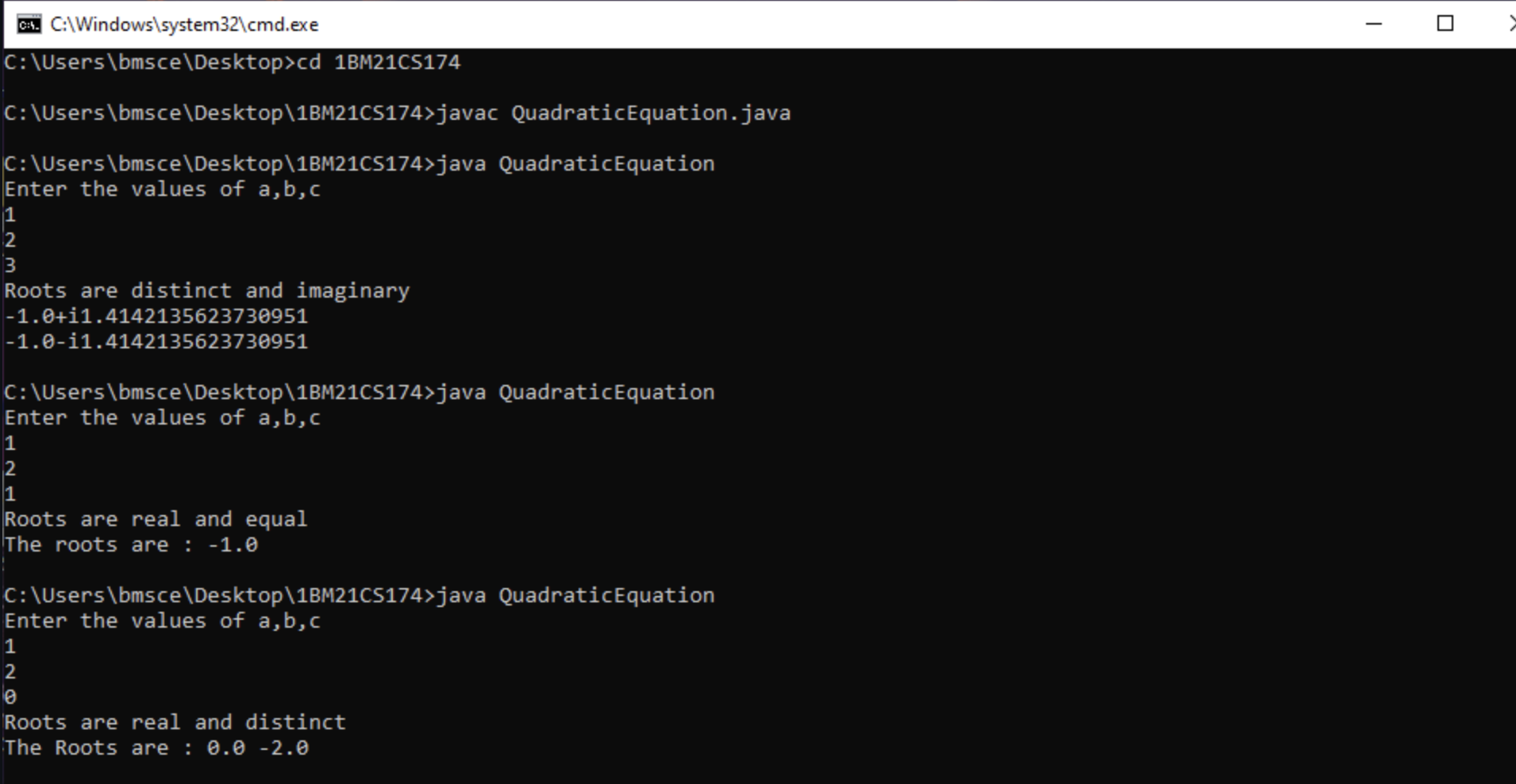
r2=(Math.sqrt(Math.abs(d)))/(2\*a);

System.out.println("Roots are distinct and imaginary ");

System.out.println(r1+"+i"+r2+"\n"+r1+"-i"+r2);

}

}

}

WEEK-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.

Code:

import java.util.Scanner;

class Student

{

String USN;

String name;

int[] credits = new int[20];

int[] marks = new int[20];

public void input(int n)

{

Scanner s = new Scanner(System.in);

System.out.print("Enter Student USN: ");

USN = s.nextLine();

System.out.print("Enter Student Name: ");

name = s.nextLine();

for(int i=0;i<n;i++)

{

System.out.print("Enter the Subject "+(i+1)+" marks and credits respectively: ");

marks[i] = s.nextInt();

credits[i] = s.nextInt();

}

}

public float calculate(int n)

{

int sum\_of\_credits = 0;

float result=0.0f;

for(int i=0;i<n;i++)

{

sum\_of\_credits+=credits[i];

if(calculate\_grade\_point(marks[i])==-1)

return -1.0f;

else

{

result = result +(float) (calculate\_grade\_point(marks[i])\*credits[i]);

}

}

return (result/sum\_of\_credits);

}

public int calculate\_grade\_point(int marks)

{

if(marks>=90)

return 10;

else if ((marks>=80)&&(marks<90))

return 9;

else if ((marks>=70)&&(marks<80))

return 8;

else if ((marks>=60)&&(marks<70))

return 7;

else if ((marks>=50)&&(marks<60))

return 6;

else if ((marks>=40)&&(marks<50))

return 5;

return -1;

}

public void display(int n,float result)

{

System.out.println("\n");

System.out.println("Student Details");

System.out.println();

System.out.println("Student USN: "+USN);

System.out.println("Student Name: "+name);

System.out.println("Student Marks and Credits");

for(int i=0;i<n;i++)

{

System.out.println("Subject " + (i+1) + "-->\tMarks: "+marks[i]+" Credits: "+credits[i]);

}

System.out.println("SGPA: "+result);

}

}

public class SGPA

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

Student s1 = new Student();

System.out.print("Enter the number of subjects: ");

int n = s.nextInt();

s1.input(n);

float result = s1.calculate(n);

if(result == -1.0f)

{

System.out.println();

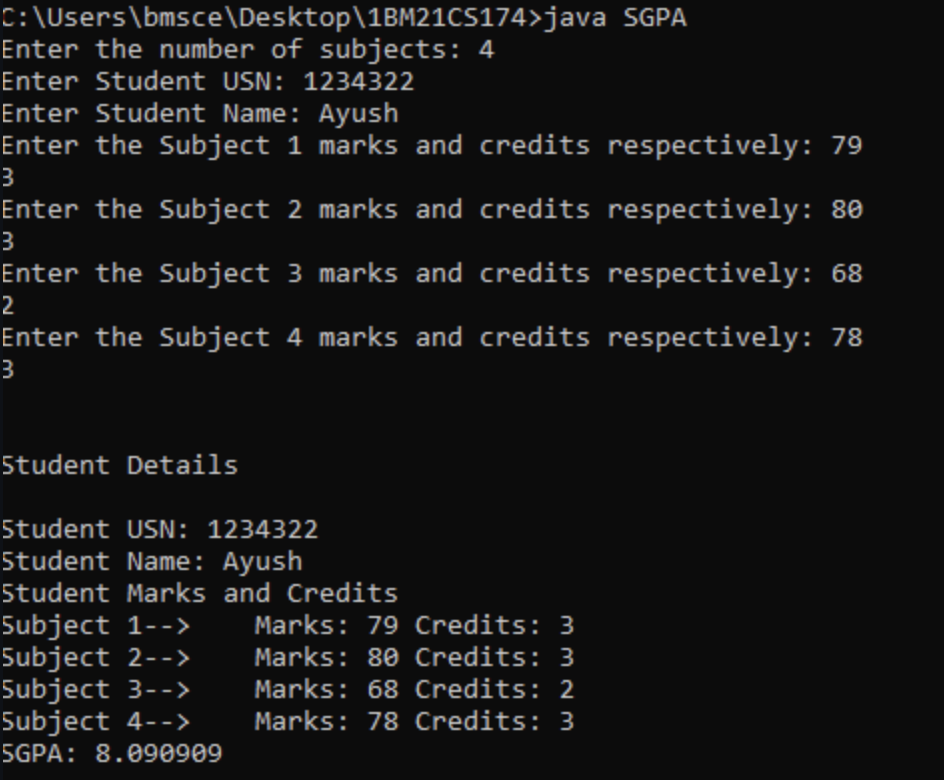
System.out.println("The Student has failed in a subject. SGPA cannot be calculated!");

System.exit(0);

}

s1.display(n,result); }}

Output:



WEEK-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.

Code:

import java.util.\*;

class Book {

String title, author;

double price;

int numPages;

Book() {

title="Default";

author="Default";

price=0.0;

numPages=0;

}

void setTitle(String t) {

title=t;

}

void setAuthor(String a) {

author=a;

}

void setPrice(double p) {

price=p;

}

void setPages(int np) {

numPages=np;

}

public String toString() {

return title+"\t"+author+"\t"+price+"\t"+numPages+"\n";

}

}

class BookDetails {

public static void main(String args[]) {

String t, a;

double p;

int np,n;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of Books");

n = sc.nextInt();

Book b[]= new Book[n];

for(int i=0; i<n;i++) {

System.out.println("Enter the Title of the Books");

t= sc.next();

System.out.println("Enter the Author of the Books");

a= sc.next();

System.out.println("Enter the Price of the Books");

p= sc.nextDouble();

System.out.println("Enter the Number of pages of the Books");

np= sc.nextInt();

b[i] = new Book();

b[i].setTitle(t);

b[i].setAuthor(a);

b[i].setPrice(p);

b[i].setPages(np);

}

System.out.println("Title \t Author \t Price \t Pages\n");

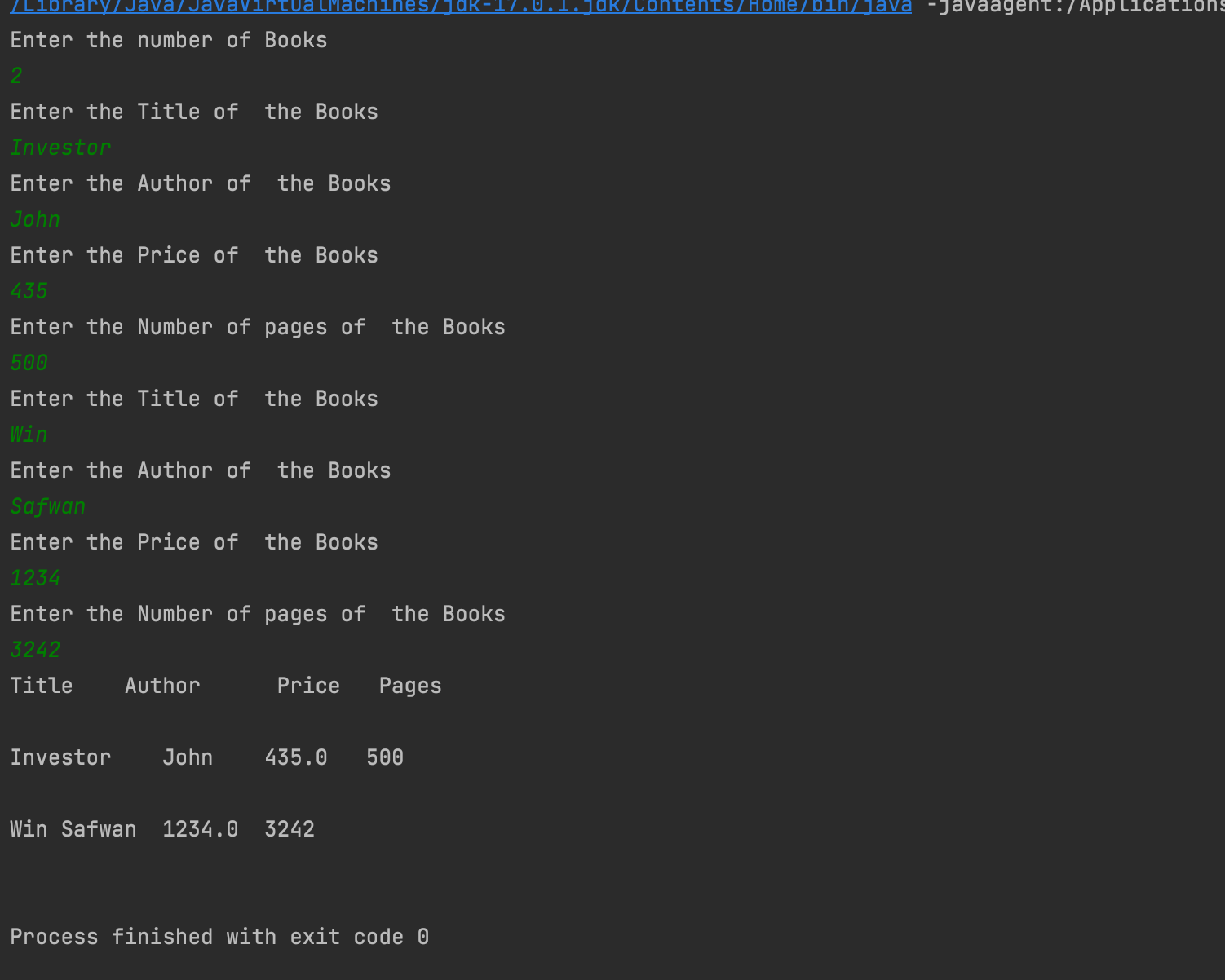
for(int i=0; i<n;i++) {

System.out.println(b[i]);

}

}

}



WEEK-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.\*;

abstract class Shape {

public int x,y;

public abstract void printArea();

}

class Rectangle1 extends Shape {

public void printArea() {

float area;

area= x \* y;

System.out.println("Area of Rectangle is " +area);

}

}

class Triangle extends Shape {

public void printArea() {

float area;

area= (x \* y) / 2.0f;

System.out.println("Area of Triangle is " + area);

}

}

class Circle extends Shape {

public void printArea() {

float area;

area=(22 \* x \* x) / 7.0f;

System.out.println("Area of Circle is " + area);

}

}

public class AreaOfShapes {

public static void main(String[] args) {

int choice;

Scanner sc=new Scanner(System.in);

System.out.println("Menu \n 1.Area of Rectangle \n 2.Area of Traingle \n 3.Area of Circle ");

System.out.print("Enter your choice : ");

choice=sc.nextInt();

switch(choice) {

case 1: System.out.println("Enter length and breadth for area of rectangle : ");

Rectangle1 r = new Rectangle1();

r.x=sc.nextInt();

r.y=sc.nextInt();

r.printArea();

break;

case 2: System.out.println("Enter bredth and height for area of traingle : ");

Triangle t = new Triangle();

t.x=sc.nextInt();

t.y=sc.nextInt();

t.printArea();

break;

case 3: System.out.println("Enter radius for area of circle : ");

Circle c = new Circle();

c.x = sc.nextInt();

c.printArea();

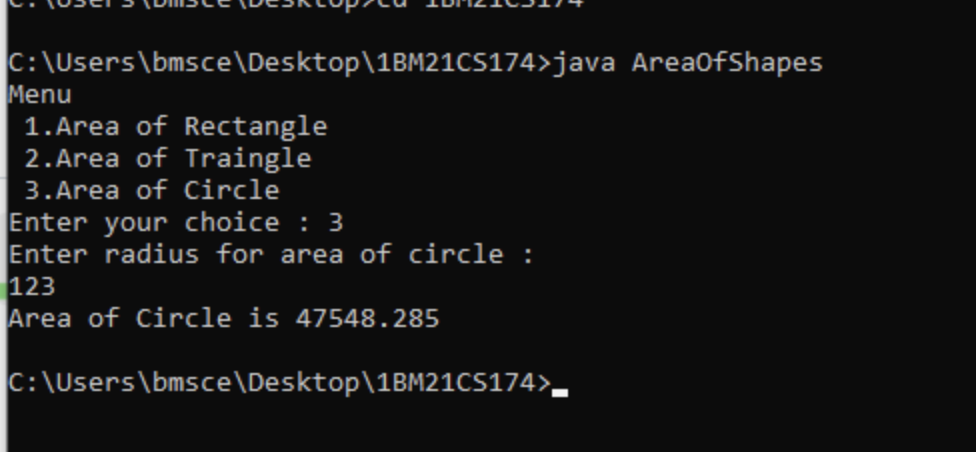
break;

default:System.out.println("Enter correct choice");

}

}

}



WEEK-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:

1. a)  Accept deposit from customer and update the balance.
2. b)  Display the balance.
3. c)  Compute and deposit interest
4. d)  Permit withdrawal and update the balance
5. Check for the minimum balance, impose penalty if necessary and update the balance.

Code:

import java.util.Scanner;

class Account

{

String customer\_name;

long acc\_no;

float bal;

Scanner s = new Scanner(System.in);

public void input()

{

System.out.print("\nEnter the Customer Name: ");

customer\_name = s.nextLine();

System.out.print("\nEnter the Account Number: ");

acc\_no = s.nextLong();

System.out.print("\nEnter the Starting Amount (Minimum Amount = 5000): ");

bal = s.nextFloat();

if(bal<5000f)

{

System.out.println("\nAccount Balance cannot be less than 5000.0 \n");

System.exit(0);

}

}

public void display()

{

System.out.println("\nCustomer Name: "+customer\_name);

System.out.println("Account Number: "+acc\_no);

System.out.println("Amount: "+bal);

}

}

class Savings extends Account

{

Scanner s = new Scanner(System.in);

float deposit,withdraw,interest;

public void deposit()

{

System.out.print("\nEnter the amount to be deposited: ");

deposit = s.nextFloat();

bal+=deposit;

System.out.println("\nBalance: "+bal);

}

public void withdraw()

{

System.out.print("\nEnter the amount to be withdrawn: ");

withdraw = s.nextFloat();

if(bal<5000)

{

System.out.println("\nInsufficient Balance");

}

else

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void check\_Bal()

{

if(bal<5000)

{

System.out.println("\nInsufficient Balance!!\nBalance: "+bal);

}

else

{

System.out.println("\nBalance: "+bal);

}

}

public void interest()

{

interest=(bal\*6)/100;

bal+=interest;

System.out.println("\nInterest Credited: "+interest+"\nBalance :"+bal) ;

}

}

class Current extends Account

{

float deposit, withdraw, penalty;

public void deposit()

{

System.out.print("\nEnter Amount to be deposited: ");

deposit = s.nextFloat();

bal += deposit;

System.out.println("Balance: " + bal);

}

public void check\_Bal()

{

if (bal < 5000)

{

penalty = (0.1f \* bal);

System.out.println("\nInitial Account Balance: "+bal);

bal = bal-penalty;

System.out.println("\nLow balance!\nPenalty Amount: " + penalty + "\nAccount balance: " + bal);

}

else

{

System.out.println("\n Balance: " + bal);

}

}

public boolean check\_Bal\_part\_2()

{

if (bal < 5000)

{

penalty = (0.1f \* bal);

System.out.println("\nInitial Account Balance: "+bal);

bal = bal-penalty;

System.out.println("\nLow Balance!\nPenalty Amount: " + penalty + "\nAccount balance: " + bal);

return false;

}

return true;

}

public void withdraw()

{

System.out.print("\nEnter Amount to withdraw: ");

withdraw = s.nextFloat();

if(check\_Bal\_part\_2())

{

bal-=withdraw;

System.out.println("\nAmount Withdrawn: "+withdraw+"\nBalance: "+bal);

}

}

public void chequebook()

{

System.out.println("\nCheque Book has been Issued!");

}

}

public class Bank

{

public static void main(String[] args)

{

Scanner s = new Scanner(System.in);

String ch;

int n;

Current c = new Current();

Savings sa = new Savings();

System.out.print("\nEnter the Account Type (S for Savings , C for Current) : ");

ch = s.next();

switch(ch.toLowerCase())

{

case "s" : sa.input();

do

{

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Check Interest"

+"\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch(n)

{

case 1 : sa.deposit();

break;

case 2 : sa.withdraw();

break;

case 3 : sa.check\_Bal();

break;

case 4 : sa.interest();

break;

case 5 : sa.display();

break;

case 6 : System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default : System.out.println("\nInvalid Operation");

}

}while(true);

case "c" : c.input();

do {

System.out.print("\n1. Deposit \n2. Withdrawal \n3. Check Balance \n4. Issue Cheque Book"

+ "\n5. Show Account Details \n6. Exit Transaction\n\nEnter your choice: ");

n = s.nextInt();

switch (n) {

case 1:

c.deposit();

break;

case 2:

c.withdraw();

break;

case 3:

c.check\_Bal();

break;

case 4:

c.chequebook();

break;

case 5:

c.display();

break;

case 6:

System.out.println("\nExiting Transaction!");

System.exit(0);

break;

default:

System.out.println("\nInvalid Operation");

}

}while(true);

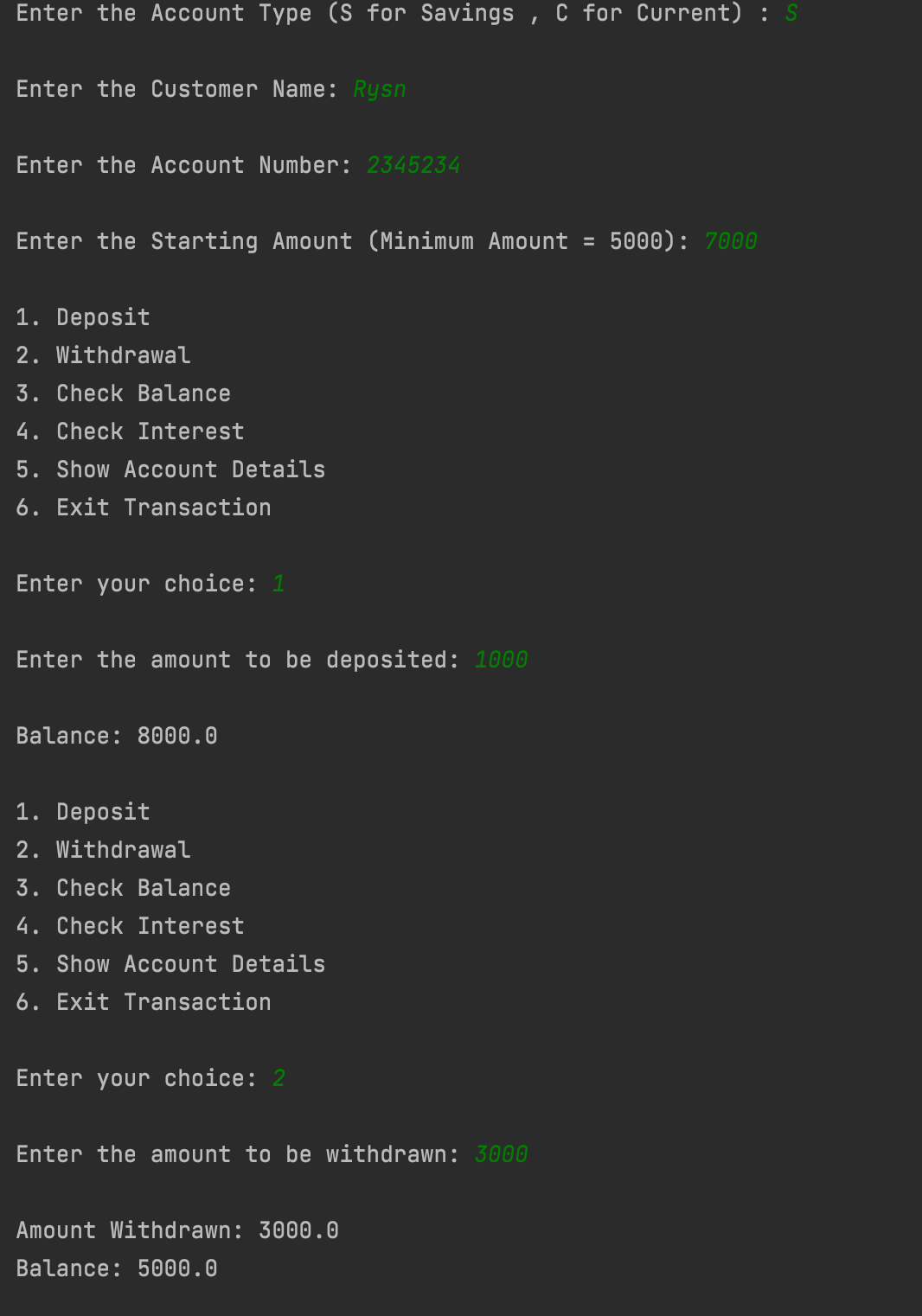
default : System.out.println("\nInvalid Choice");

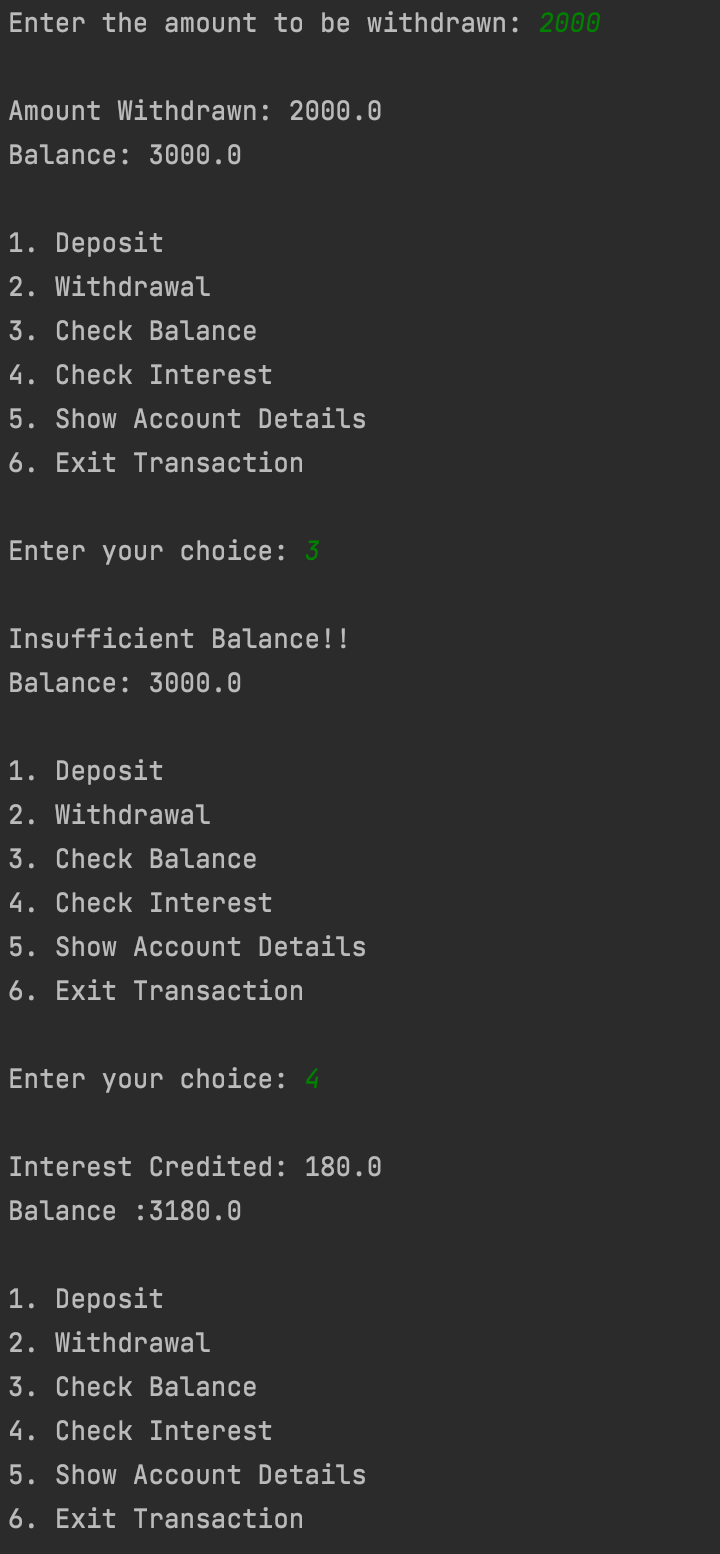
break;

}

}

}





WEEK-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.

Code:

import java.util.\*;

class Father extends Exception

{

int father\_age;

Father(int Fage)

{

father\_age=Fage;

}

public String toString()

{

return "Wrong age";

}

}

class Son extends Father

{

int son\_age;

Son(int Fage,int Sage)

{

super(Fage);

son\_age=Sage;

}

public String toString()

{

return " Invalid son's age is greater than or equal to Father's age";

}

}

class Main

{

static int a,b;

static void fatherage(int a) throws Father

{

System.out.println("Called father's age is: "+a);

if(a<0)

throw new Father(a);

System.out.println("No exception in Father's age:"+a);

}

static void sonage(int a,int b) throws Son

{

System.out.println("Called son's age is: "+b);

if(b>=a)

throw new Son(a,b);

System.out.println("No exception in Son's age:"+b);

}

public static void main(String args[])

{

Scanner age=new Scanner(System.in);

System.out.println("Enter Father's age:");

a=age.nextInt();

System.out.println("Enter Son's age:");

b=age.nextInt();

try

{

fatherage(a);

}

catch(Father e)

{

System.out.println(e);

}

try

{

sonage(a,b);

}

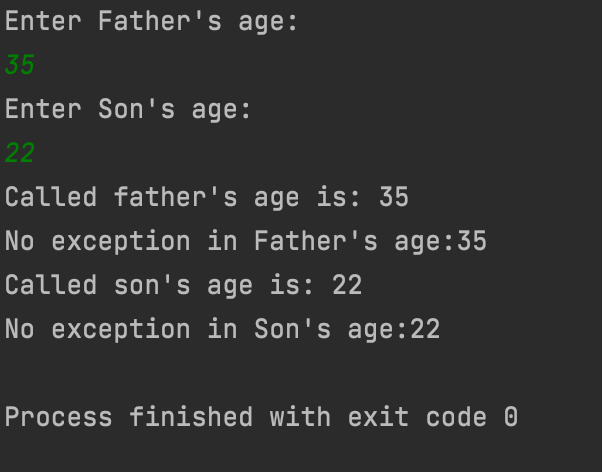
catch(Son e)

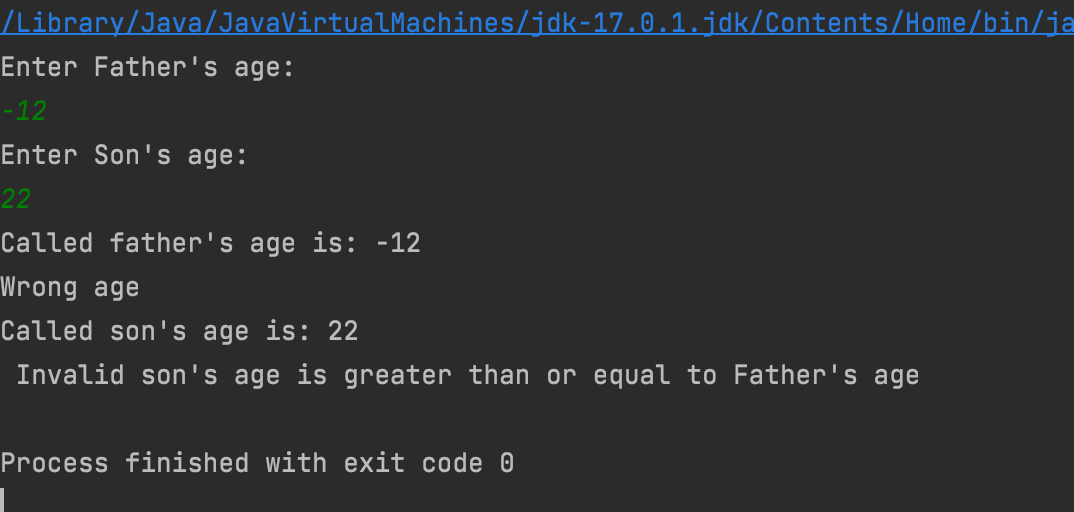
{

System.out.println(e);

}

}

}



WEEK-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.\*;

class Thread\_1 extends Thread

{

public void run()

{

int i = 0;

while(i<100)

{

try

{

Thread.sleep(10000);

System.out.println("BMSCE");

}

catch(Exception e)

{

System.out.println("Exception: "+e);

}

i++;

}

}

}

class Thread\_2 extends Thread

{

public void run()

{

int i = 0;

while(i<100)

{

try

{

Thread.sleep(2000);

System.out.println("CSE");

}

catch(Exception e)

{

System.out.println("Exception "+e);

}

i++;

}

}

}

public class Threads

{

public static void main(String[] args)

{

Thread t1 = new Thread\_1();

Thread t2 = new Thread\_2();

t1.start();

t2.start();

}

}

