OOJ Lab Record

LAB 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.

message stating that there are no real solutions.

import java.util.Scanner;

public class Quadratic

{

public static void main(String[] args) {

double a,b,c,d;

double root1,root2;

Scanner inp =new Scanner(System.in);

System.out.println("enter the value of a,b and c

respectively:");

a=inp.nextDouble();

b=inp.nextDouble();

c=inp.nextDouble();

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

if(d>0){

System.out.println("roots are real");

root1= (-b + Math.sqrt(d))/(2 \* a);

root2= (-b - Math.sqrt(d))/(2 \* a);

System.out.println("root1="+root1+" roots="+root2);

}

else if(d==0){

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

root1 = root2 = -b / (2 \* a);

}

else{

System.out.println("roots are imaginary");

}

}

}

Output



LAB 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{

String name;

int i,n,usn,grade=0;

int marks[]=new int [5];

int credits[]=new int[6];

double total=0;

void get\_input()

{

Scanner m= new Scanner(System.in);

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

name=m.next();

System.out.println("Enter USN of student:");

usn=m.nextInt();

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

n=m.nextInt();

System.out.println("Enter the credits and marks of suject:");

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

{

System.out.println("Enter marks of Credits "+(i+1)+" :");

credits[i]=m.nextInt();

System.out.println("Enter marks of subject"+(i+1)+":");

marks[i]=m.nextInt();

}

}

void calculate()

{

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

{

if(marks[i]>=90 && marks[i]<=100)

grade=10;

else if(marks[i]>=80 && marks[i]<=90)

grade=9;

else if(marks[i]>=70 && marks[i]<=80)

grade=8;

else if(marks[i]>=60 && marks[i]<=70)

grade=7;

else if(marks[i]>=50 && marks[i]<=60)

grade=6;

else if(marks[i]>=40 && marks[i]<=50)

grade=5;

else if(marks[i]>=0 && marks[i]<=40)

grade=0;

else

System.out.println("invalid");

total=total+(grade\*credits[i]);

}

total=total/20;

System.out.println("Sgpa="+total);

}

void display\_output()

{

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

System.out.println("usn"+usn);

System.out.println("Sgpa="+total);

}

public static void main(String args[])

{

Student s1=new Student();

s1.get\_input();

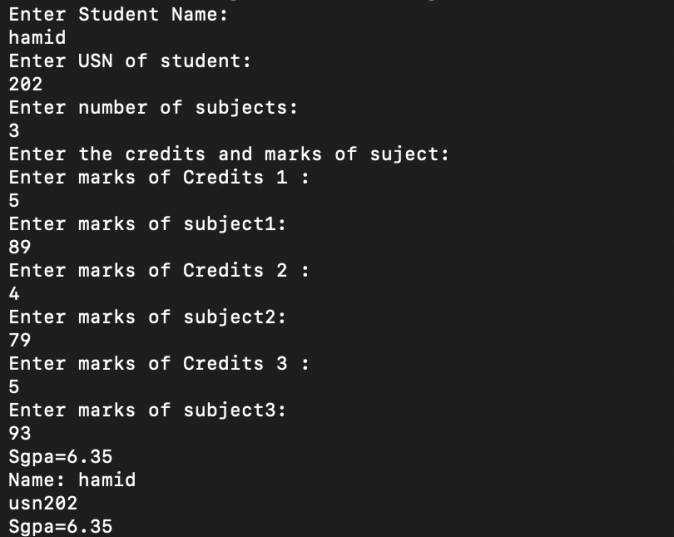
s1.calculate();

s1.display\_output();

}

}

Output



LAB 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.\*;

class Book

{

private String name;

private String author;

private double price;

private int num\_pages;

Book()

{

name="xyz";

author="abc";

price= 0.0;

num\_pages=10;

}

void getdata()

{

Scanner sc= new Scanner(System.in);

System.out.println("enter the name of book");

name = sc.nextLine();

System.out.println("enter the name of author");

author = sc.nextLine();

System.out.println("enter the price of book");

price = sc.nextDouble();

System.out.println("enter the number of pages");

num\_pages = sc.nextInt();

}

public String toString()

{

return("Book: "+name+"\nAuthor: "+author+"\nPrice: Rs "+price+"\nNo.of pages: "+num\_pages);

}

}

class lab4

{

public static void main(String ss[])

{

Scanner xx=new Scanner(System.in);

System.out.println("Enter the no of books:");

int n=xx.nextInt();

Book b[]=new Book[n];

int i;

System.out.println("Enter the book details");

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

{

System.out.println("Book "+(i+1));

b[i]=new Book();

b[i].getdata();

}

System.out.println("Printing book details....");

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

{

System.out.println("Book "+(i+1));

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

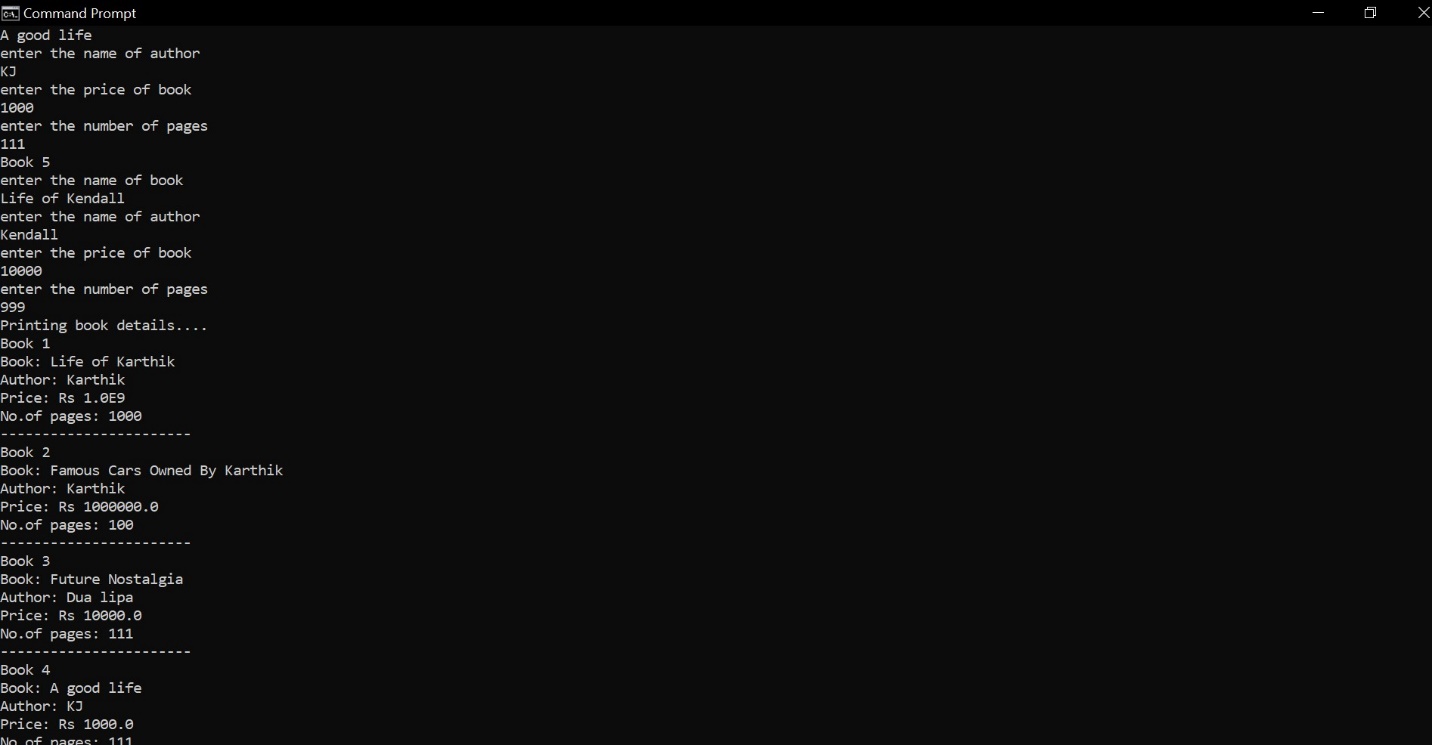
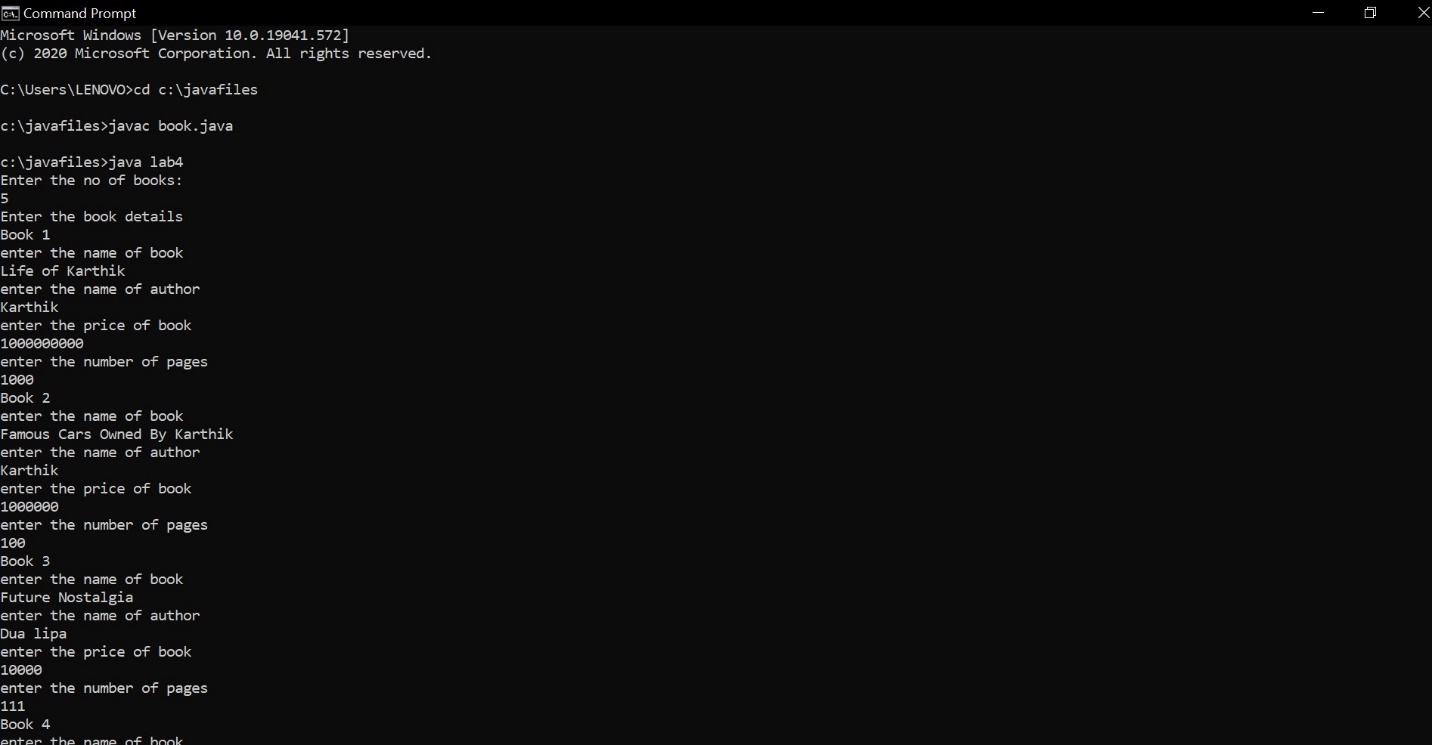
System.out.println("-----------------------");

}

}

}

Output



Lab 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.

import java.util.\*;

abstract class Shape

{

int a;

int b;

abstract void printArea();

}

class Rectangle extends Shape

{

Rectangle(int x, int y)

{

a=x;

b=y;

}

void printArea()

{

System.out.println("Area is "+(a\*b));

}

}

class Triangle extends Shape

{

Triangle(int x, int y)

{

a=x;

b=y;

}

void printArea()

{

System.out.println("Area is "+(a\*b\*0.5));

}

}

class Circle extends Shape

{

Circle(int x)

{

a=x;

}

void printArea()

{

System.out.println("Area is "+(a\*a\*3.14));

}

}

class lab5

{

public static void main(String ss[])

{

int l,b,ba,h,ra;

Scanner sc = new Scanner(System.in);

System.out.println("enter the length and breadth of rectangle");

l= sc.nextInt();

b= sc.nextInt();

Rectangle r= new Rectangle(l,b);

r.printArea();

System.out.println("enter the base and height of triangle");

ba= sc.nextInt();

h= sc.nextInt();

Triangle t = new Triangle(ba,h);

t.printArea();

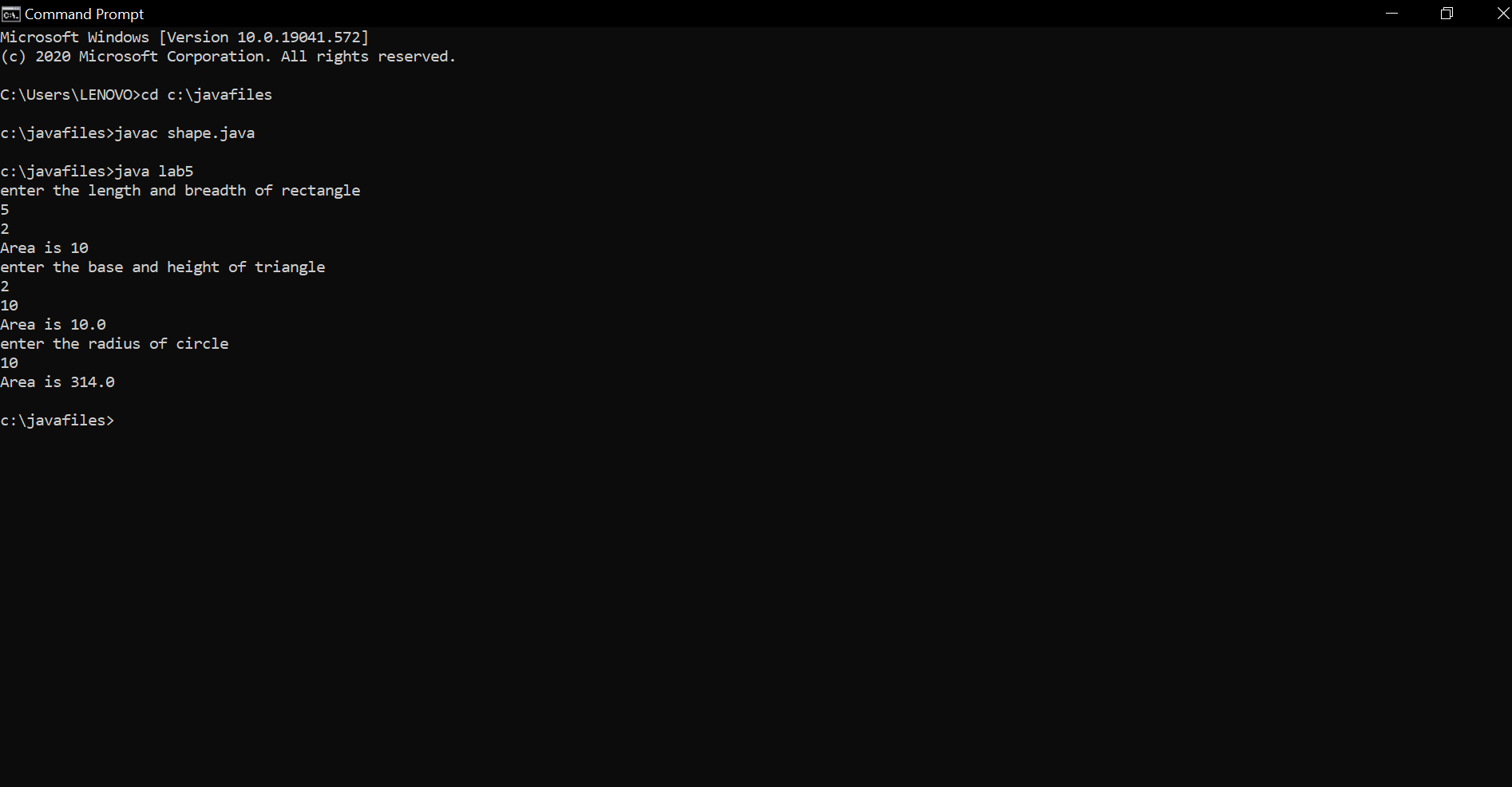
System.out.println("enter the radius of circle");

ra= sc.nextInt();

Circle c = new Circle(ra);

c.printArea();

}

}

Lab 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 Curr-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.

import java.util.\*;

import java.lang.Math;

class Account

{

String name;

int acctno;

char type;

double balance;

double dep;

boolean cheq;

void get(char c)

{

type = c;

if(c=='s' || c == 'S')

cheq=false;

else cheq=true;

Scanner sc = new Scanner(System.in);

System.out.println("Enter your name");

name = sc.nextLine();

System.out.println("Enter the account number");

acctno = sc.nextInt();

System.out.println("Enter the current available balance in your account");

balance= sc.nextDouble();

}

void putd()

{

System.out.println("Account details");

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

System.out.println("Account number: "+acctno);

System.out.println("Account type :"+type);

System.out.println("balance: "+balance);

}

void dep()

{

Scanner ss = new Scanner(System.in);

System.out.println("Enter the amount to be deposited");

dep= ss.nextDouble();

balance=balance +dep;

System.out.println("Amount has been deposited and balance has been updated");

}

void display()

{

System.out.println("Balance amount is "+balance);

}

void check()

{

if(cheq==false)

System.out.println("Cheque book facility is not available");

else

System.out.println("Cheque book facility is available");

}

}

class Saving extends Account

{

double rate;

double s\_with;

int n;

int ch;

double amt;

double term;

double pr;

void ci()

{

Scanner ss = new Scanner(System.in);

System.out.println("Enter principal deposit amount");

pr = ss.nextDouble();

System.out.println("Enter the rate of interest");

rate = ss.nextDouble();

System.out.println("Enter the term(years)");

term = ss.nextDouble();

System.out.println("Enter the number of times interest in compounded annually");

n = ss.nextInt();

amt = pr\* Math.pow((1+(rate/100)),(n\*term));

balance+= amt;

System.out.println("Interest is compounded and deposited; balance is updated");

}

void with\_s()

{

Scanner ss = new Scanner(System.in);

System.out.println("Enter the amount of money to be withdrawn");

s\_with = ss.nextDouble();

if(s\_with>balance)

System.out.println("Insufficient balance");

else

{balance= balance - s\_with;

System.out.println("Money has been withdrawn and balance has been updated");}

}

}

class Current extends Account

{

double c\_with;

double pen;

double min;

Current()

{

pen=100;

min=500;

}

void with\_c()

{

Scanner xx = new Scanner(System.in);

System.out.println("Enter the amount to be withdrawn");

c\_with= xx.nextDouble();

if(c\_with>balance)

{System.out.println("Insufficient funds!");

return;}

else

{balance= balance- c\_with;

System.out.println("Amount has been withdrawn and balance has been updated");}

if(balance<min)

{

System.out.println("Balance is below the minimum threshold. Service penalty charge = 100/- .");

if(balance<pen)

System.out.println("Due to insufficient funds, penalty charge will be deducted from account after replenishing. Current balance is "+balance);

else

{

balance= balance-pen;

System.out.println("Penalty charge has been deducted from account balance. Current balance is "+balance);

}

}

}

}

class lab4

{

public static void main(String sss[])

{

int cch, chh;

Scanner sx = new Scanner(System.in);

System.out.println("--------Welcome----------");

System.out.println("Savings account or current account? 1- Savings; 2- Current");

int ch= sx.nextInt();

if(ch==1)

{

Saving s = new Saving();

s.get('S');

do{

System.out.println("1. Deposit money\n2. Calculate compound interest\n3. Withdraw money\n4. Display balance\n5. Cheque book facility\n6. Exit");

System.out.println("Enter your choice");

chh= sx.nextInt();

switch(chh)

{

case 1:

s.dep();

break;

case 2:

s.ci();

break;

case 3:

s.with\_s();

break;

case 4:

s.display();

break;

case 5:

s.check();

break;

case 6:

break;

default:

System.out.println("Wrong option.");

break;

}

}while(chh!=6);

}

else if(ch==2)

{

Current cr = new Current();

cr.get('C');

do{

System.out.println("1. Deposit money\n2. Chequebook facility\n3. Withdraw money\n4. Display balance\n5. Exit");

cch= sx.nextInt();

switch(cch)

{

case 1:

cr.dep();

break;

case 2:

cr.check();

break;

case 3:

cr.with\_c();

break;

case 4:

cr.display();

break;

case 5:

break;

default:

System.out.println("Wrong option.");

break;

}

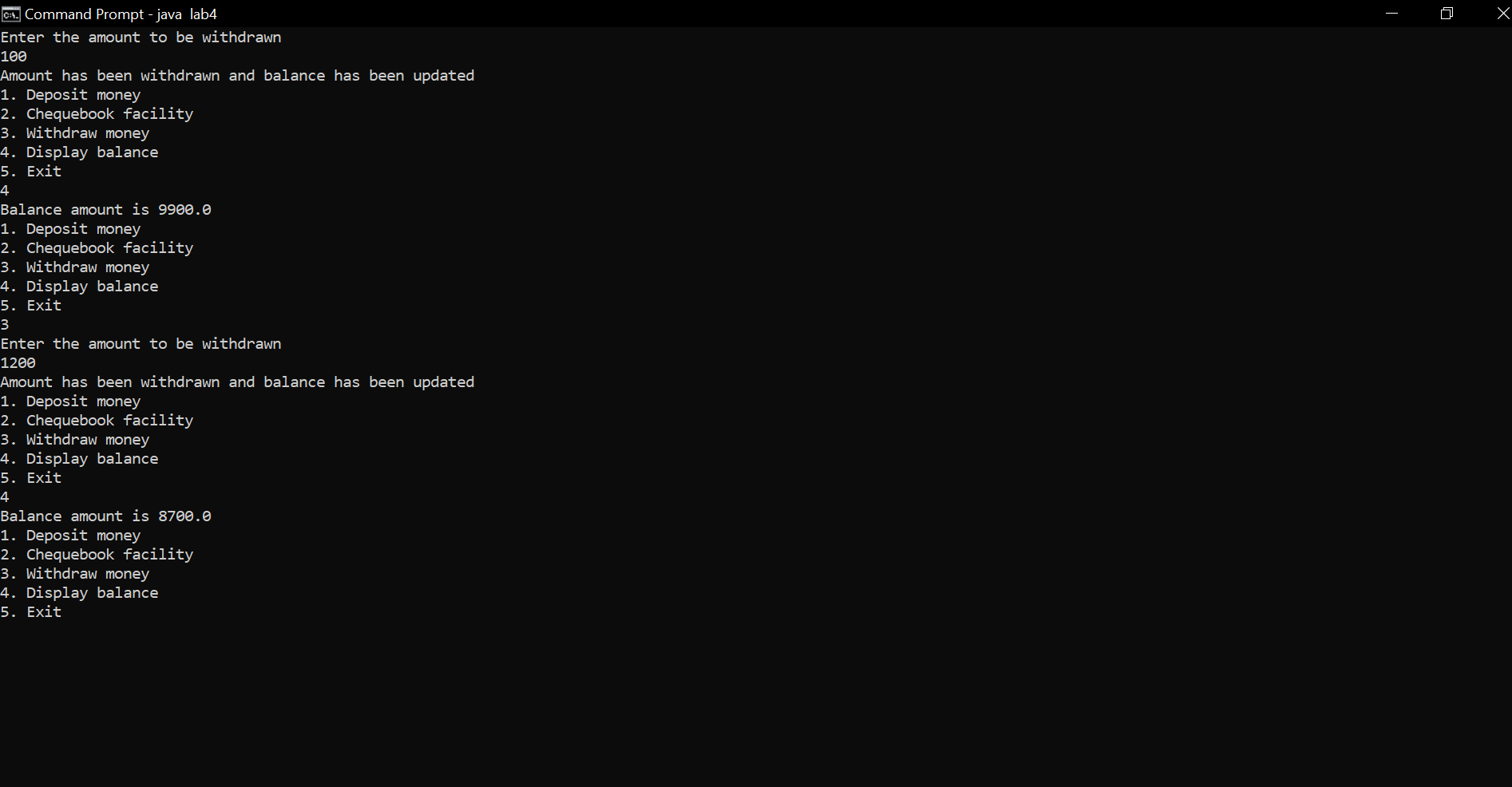
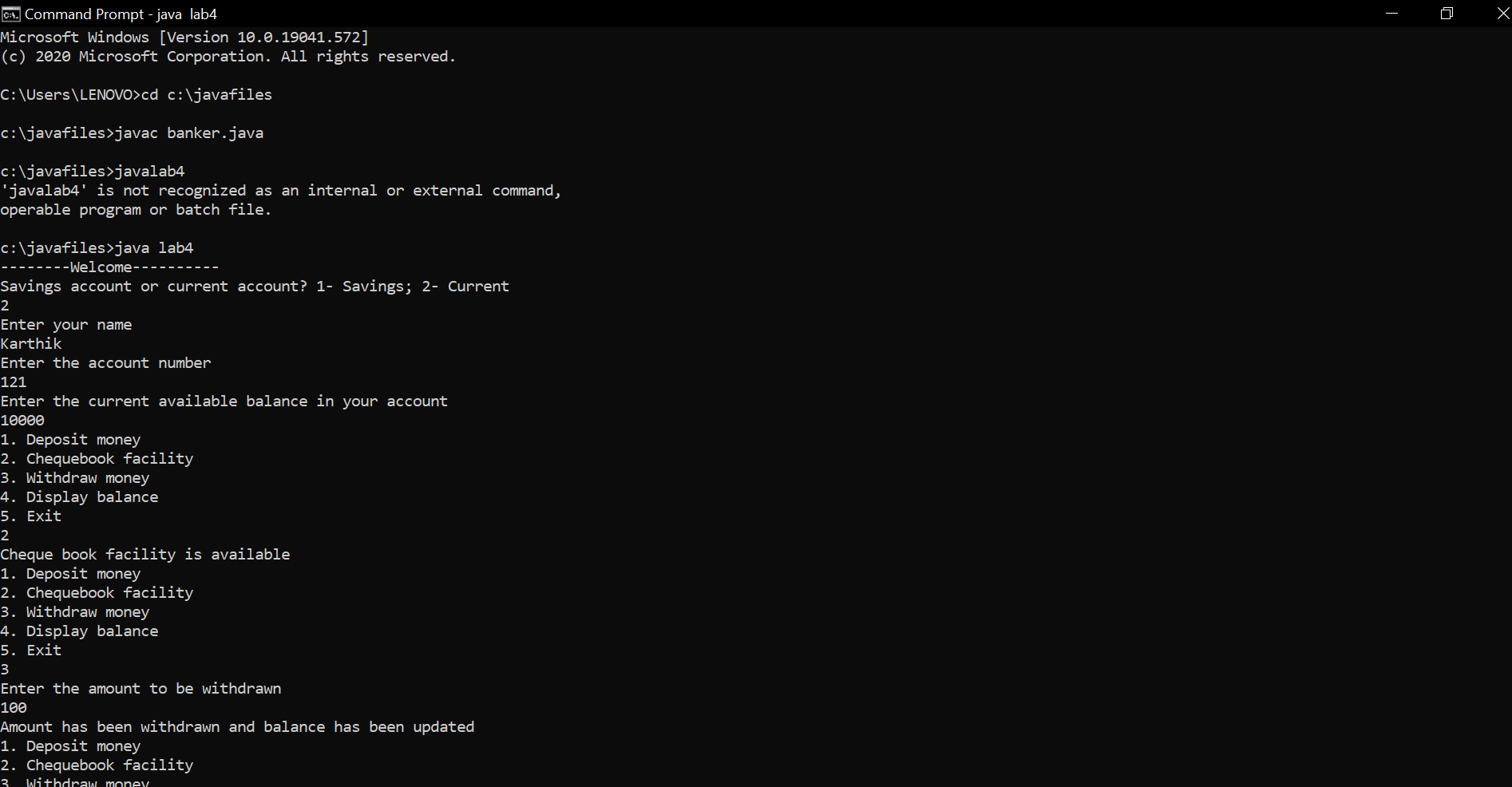
}while(cch!=5);

}

else System.out.println("Wrong!");

}

}



LAB 6:

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

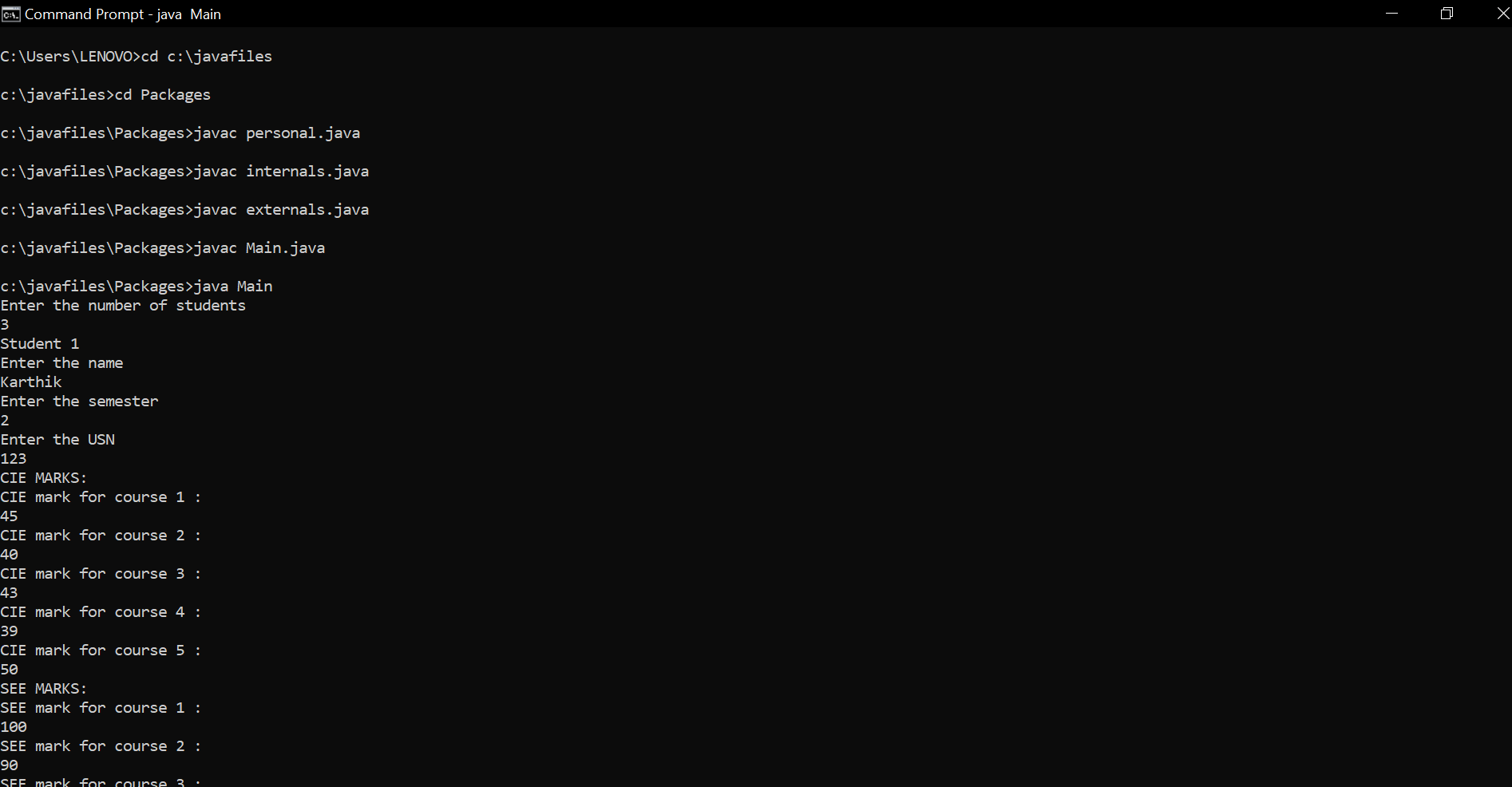
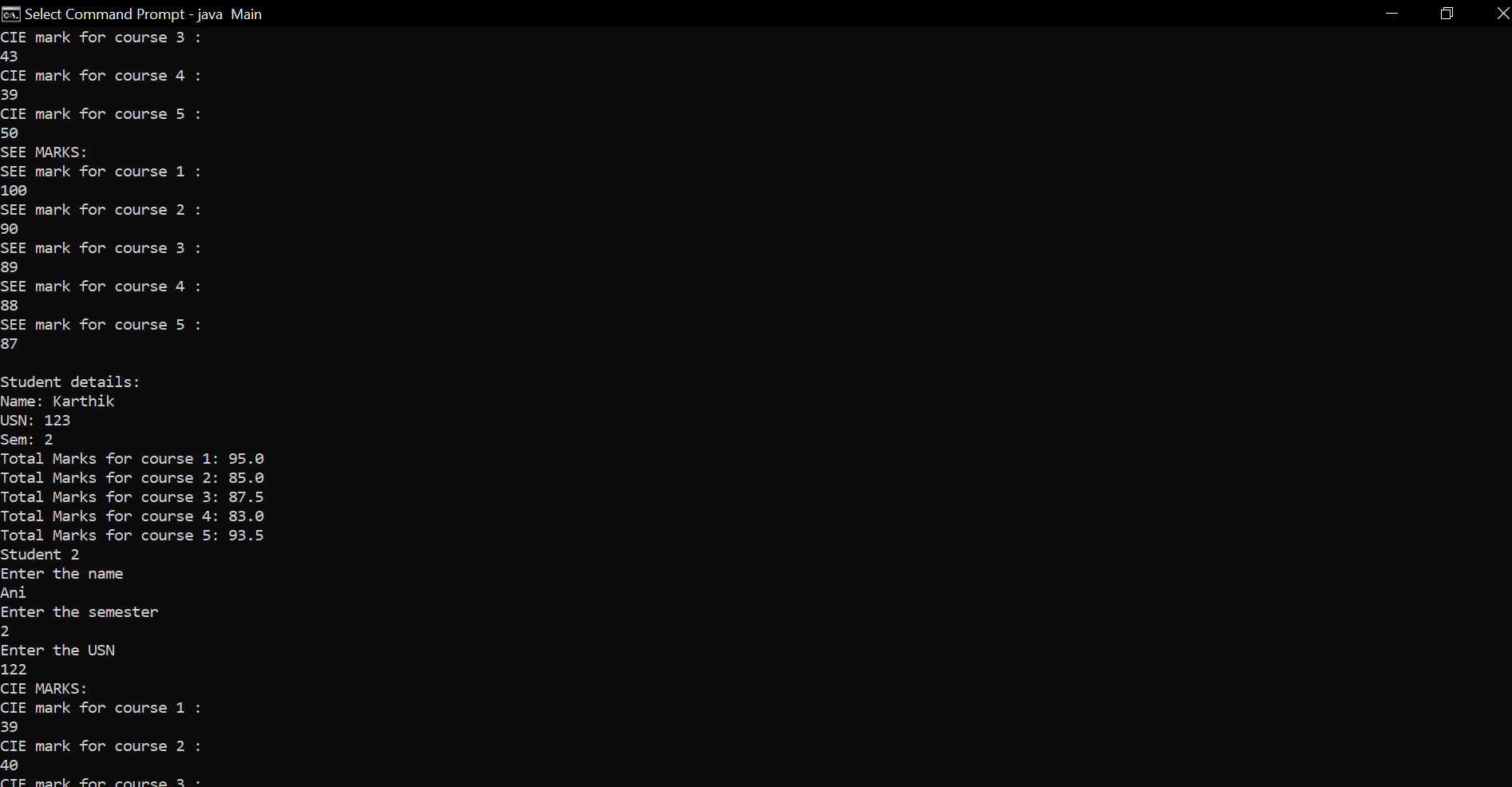
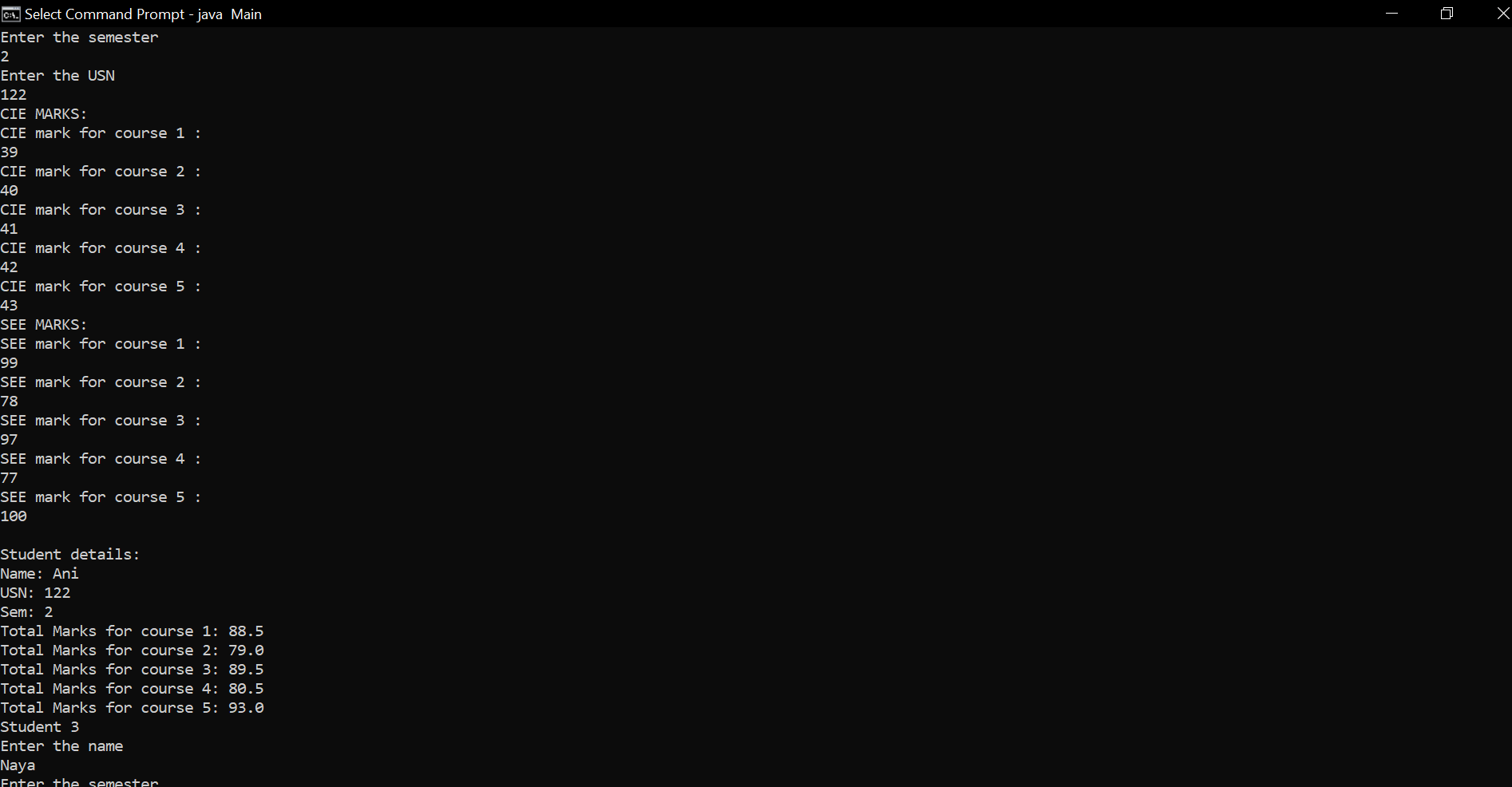
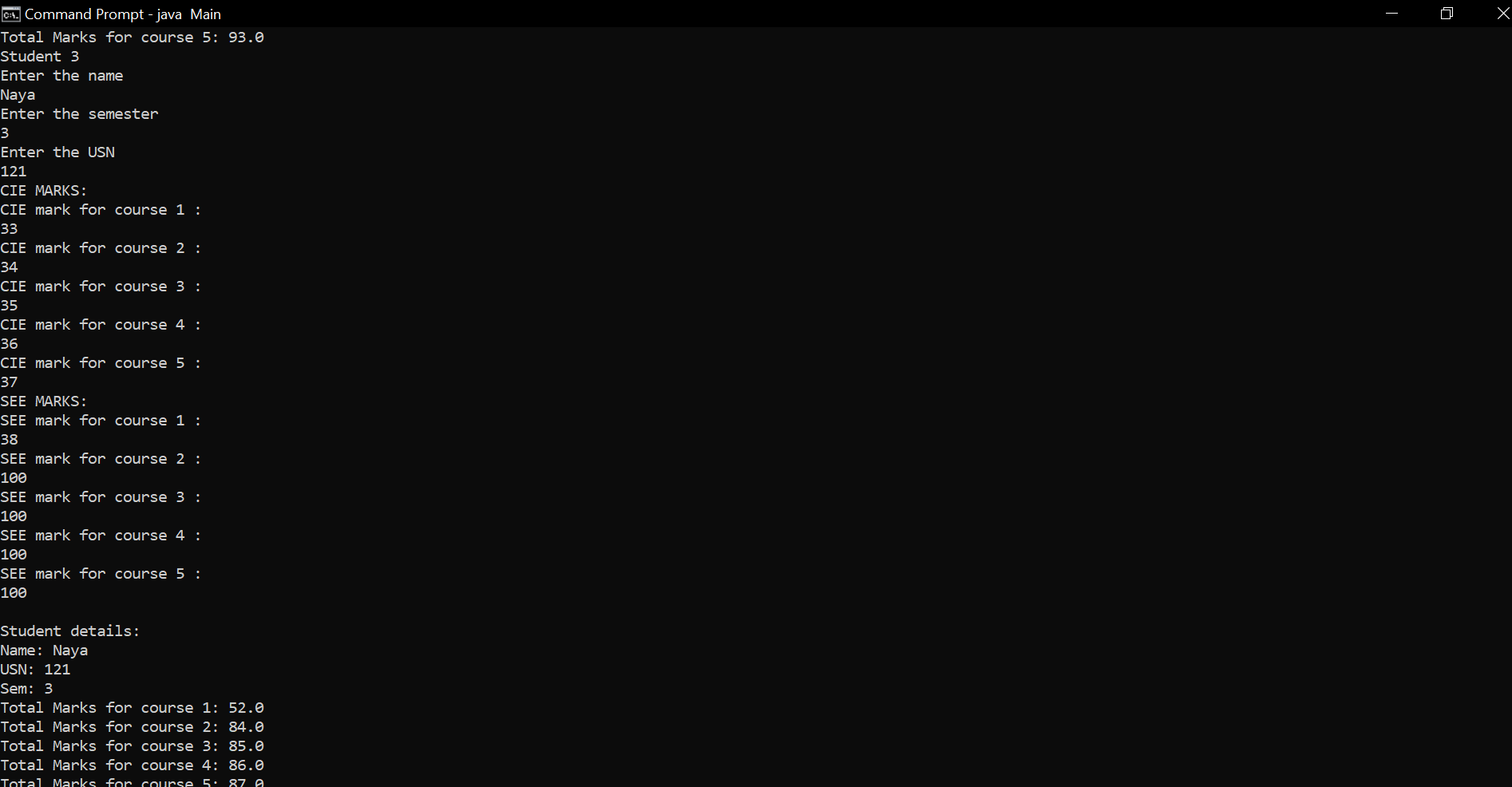
students in all five courses.

|  |
| --- |
| import CIE.\*; |
|  | import SEE.\*; |
|  | import java.util.\*; |
|  |  |
|  | class Main |
|  | { |
|  | public static void main(String args[]) |
|  | { |
|  | Scanner sx = new Scanner(System.in); |
|  | System.out.println("Enter the number of students"); |
|  | int n= sx.nextInt(); |
|  | CIE.internals in[]= new CIE.internals[n]; |
|  | SEE.externals en[]= new SEE.externals[n]; |
|  | int i,j; |
|  | for(i=0;i<n;i++) |
|  | { |
|  | System.out.println("Student "+(i+1)); |
|  | in[i] = new CIE.internals(); |
|  | en[i] = new SEE.externals(); |
|  | in[i].read(); |
|  |  |
|  | System.out.println("CIE MARKS:"); |
|  | in[i].accept(); |
|  | System.out.println("SEE MARKS:"); |
|  | en[i].get(); |
|  | System.out.println(); |
|  | in[i].display(); |
|  | for(j=0;j<5;j++) |
|  |  |
|  | System.out.println("Total Marks for course "+(j+1)+": "+(in[i].cie[j] + (en[i].see[j]/2))); |
|  | } |
|  | } |
|  | } |

|  |
| --- |
| package SEE; |
|  | import java.util.\*; |
|  | import CIE.\*; |
|  | public class externals extends personal |
|  | { |
|  | public double see[]; |
|  |  |
|  | public void get() |
|  | { |
|  | see= new double[5]; |
|  | Scanner sc = new Scanner(System.in); |
|  | for(int i=0;i<5;i++) |
|  | { |
|  | System.out.println("SEE mark for course "+(i+1)+" : "); |
|  | see[i]= sc.nextDouble(); |
|  | } |
|  | } |
|  |  |
|  |  |
|  | } |

|  |  |
| --- | --- |
|  | package CIE; |
|  | import java.util.\*; |
|  | public class internals extends personal |
|  | { |
|  | public double cie[]; |
|  |  |
|  | public void accept() |
|  | { |
|  | cie= new double[5]; |
|  | Scanner sc = new Scanner(System.in); |
|  | for(int i=0;i<5;i++) |
|  | { |
|  | System.out.println("CIE mark for course "+(i+1)+" : "); |
|  | cie[i]= sc.nextDouble(); |
|  | } |
|  | } |
|  |  |
|  |  |
|  |  |
|  | } |
|  | package CIE; |
|  | import java.util.\*; |
|  | public class personal |
|  | { |
|  | public String name; |
|  | public int sem; |
|  | public String usn; |
|  |  |
|  | public void read() |
|  | { |
|  | Scanner sc = new Scanner(System.in); |
|  | System.out.println("Enter the name"); |
|  | name = sc.next(); |
|  | System.out.println("Enter the semester"); |
|  | sem = sc.nextInt(); |
|  | System.out.println("Enter the USN"); |
|  | usn = sc.next(); |
|  | } |
|  | public void display() |
|  | { |
|  | System.out.println("Student details: "); |
|  | System.out.println("Name: "+name+"\nUSN: "+usn+"\nSem: "+sem); |
|  | } |
|  |  |
|  |  |
|  | } |

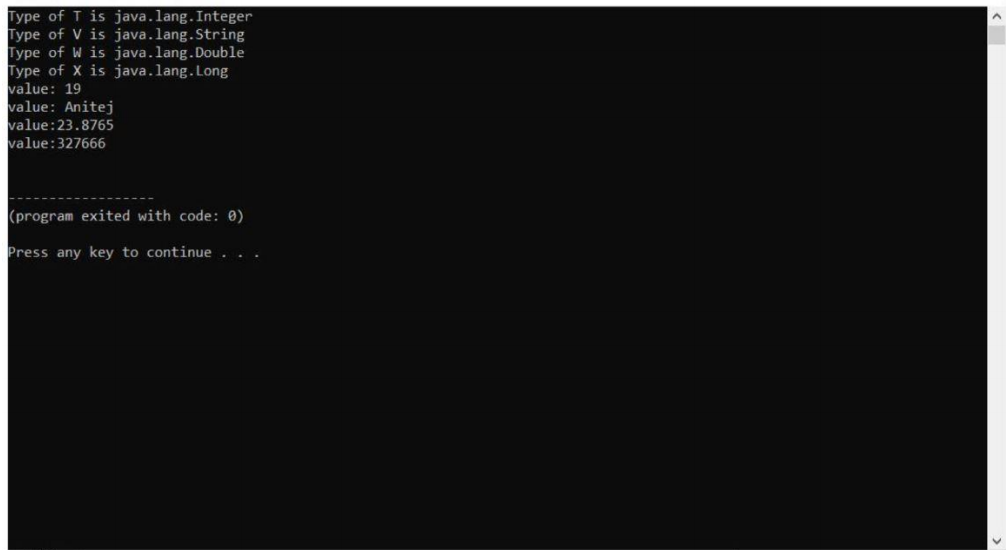
Output



Lab 7

Write a program to demonstrate generics with multiple object parameters.

|  |  |  |
| --- | --- | --- |
|  | class FourGen<T, V, W, X> { | |
|  | T ob1; | |
|  | V ob2; | |
|  | W ob3; | |
|  | X ob4; | |
|  | FourGen(T o1, V o2, W o3, X o4) { | |
|  | ob1 = o1; | |
|  | ob2 = o2; | |
|  | ob3 = o3; | |
|  | ob4 = o4; | |
|  | } | |
|  | void showTypes() { | |
|  | System.out.println("Type of T is " +ob1.getClass().getName()); | |
|  | System.out.println("Type of V is " +ob2.getClass().getName()); | |
|  | System.out.println("Type of W is " +ob3.getClass().getName()); | |
|  | System.out.println("Type of X is " +ob4.getClass().getName()); | |
|  |  | |
|  | } | |
|  | T getob1() { | |
|  | return ob1; | |
|  | } | |
|  | V getob2() { | |
|  | return ob2; | |
|  | } | |
|  | W getob3() { | |
|  | return ob3; | |
|  | } | |
|  | X getob4() { | |
|  | return ob4; | |
|  | } | |
|  | } | |
|  | class SimpGen { | |
|  | public static void main(String args[]) { | |
|  | FourGen<Integer, String, Double, Long> tgObj = | |
|  | new FourGen<Integer, String, Double, Long>(21, "Karthik", 23.8765, 327666L); | |
|  | tgObj.showTypes(); | |
|  | int v = tgObj.getob1(); | |
|  | System.out.println("value: " + v); | |
|  | String str = tgObj.getob2(); | |
|  | System.out.println("value: " + str); | |
|  | Double dbl= tgObj.getob3(); | |
|  | System.out.println("value:" + dbl); | |
|  | Long lng= tgObj.getob4(); | |
|  | System.out.println("value:" + lng); | |
|  |  | |
|  | } | |
| }  Output | |



Lab 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=father’s age.

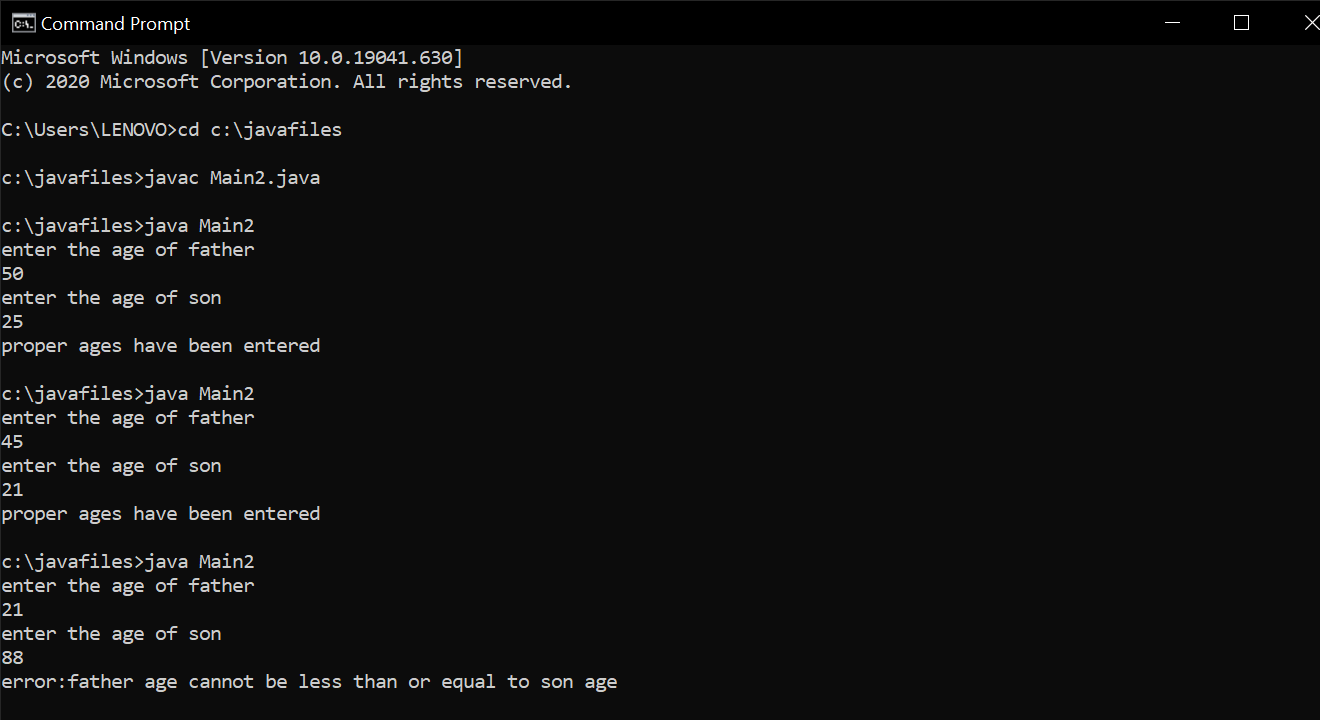
Top of Form

Bottom of Form

Top of Form

Bottom of Form

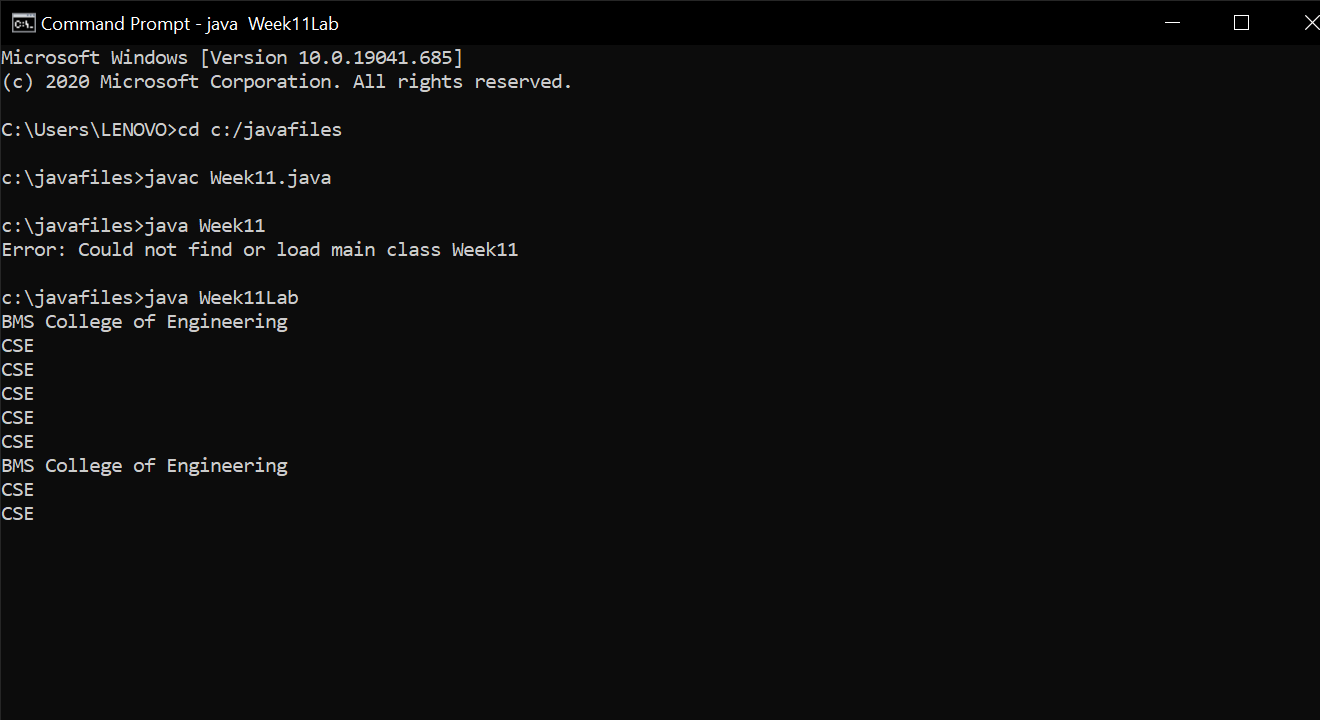
|  |  |
| --- | --- |
|  | import java.util.\*; |
|  | class WrongAge extends Exception |
|  | { |
|  | private int a1,b1; |
|  | WrongAge(int a,int b) |
|  | { |
|  | a1=a; |
|  | b1=b; |
|  | } |
|  | public String toString() |
|  | { |
|  | if(a1<0||b1<0) |
|  | return "input age cannot be less than 0"; |
|  | else if(a1<=b1) |
|  | return "father age cannot be less than or equal to son age "; |
|  | return ""; |
|  | } |
|  | } |
|  |  |
|  | class Father |
|  | { |
|  | int fage,sage; |
|  | Scanner sc=new Scanner(System.in); |
|  | Father() throws WrongAge |
|  | { |
|  | System.out.println("enter the age of father"); |
|  | fage=sc.nextInt(); |
|  | System.out.println("enter the age of son"); |
|  | sage=sc.nextInt(); |
|  | if(fage<0||sage<0) |
|  | throw new WrongAge(fage,sage); |
|  |  |
|  |  |
|  |  |
|  | } |
|  | } |
|  |  |
|  | class Son extends Father |
|  | { |
|  |  |
|  | Son() throws WrongAge |
|  | { |
|  |  |
|  | if(sage>=fage) |
|  | throw new WrongAge(fage,sage); |
|  | else |
|  | System.out.println("proper ages have been entered"); |
|  | } |
|  | } |
|  |  |
|  | class Main2 |
|  | { |
|  | public static void main(String args[]) |
|  | { |
|  |  |
|  | try |
|  | { |
|  | Son s=new Son(); |
|  | }catch(WrongAge e){ |
|  | System.out.println("error:"+e); |
|  |  |
|  | } |
|  | } |
|  | } |



Lab 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.

|  |
| --- |
| Class NewThread implements Runnable |
|  | { Thread t; |
|  | NewThread() |
|  | { |
|  | t = new Thread(this, "NThread"); |
|  | t.start(); |
|  | } |
|  |  |
|  | public void run() |
|  | { |
|  | try |
|  | { |
|  | for(int n=100;n>0;n--) |
|  | { |
|  | System.out.println("CSE"); |
|  | Thread.sleep(2000); |
|  | } |
|  | } |
|  | catch(InterruptedException ie) |
|  | { |
|  | System.out.println("Child Thread Interrupted"); |
|  | } |
|  |  |
|  | } |
|  | } |
|  |  |
|  | class Week9 |
|  | { |
|  | public static void main(String ss[]) |
|  | { |
|  | NewThread n1=new NewThread(); |
|  | //n1.t.start(); |
|  |  |
|  |  |
|  | try |
|  | { |
|  | for(int n=5;n>0;n--) |
|  | { |
|  | System.out.println("BMS College of Engineering"); |
|  | Thread.sleep(10000); |
|  | } |
|  | } |
|  | catch(InterruptedException ie) |
|  | { |
|  | System.out.println("Main Thread interrupted"); |
|  | } |
|  |  |
|  | } |
|  | } |



LAB10

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

import java.awt.\*;

import java.awt.event.\*;

class SampleDialog extends Dialog implements ActionListener {

lab12 bld;

SampleDialog(Frame parent, String title) {

super(parent, title, false);

bld=(lab12)parent;

setLayout(new FlowLayout());

setSize(300, 200);

add(new Label(bld.msg1));

Button b;

add(b = new Button("OK"));

b.addActionListener(this);

}

public void actionPerformed(ActionEvent ae) {

dispose();

}

}

public class lab12 extends Frame implements ActionListener

{

TextField num1,num2,result;

String msg="",msg1="";

Button divide;

public lab12()

{

setLayout(new FlowLayout());

Label nnum1=new Label("Num1: ",Label.RIGHT);

Label nnum2=new Label("Num2: ",Label.RIGHT);

Label rresult=new Label("Result: ",Label.RIGHT);

Button b=new Button("divide");

num1=new TextField(8);

num2=new TextField(8);

result=new TextField(8);

add(nnum1);

add(num1);

add(nnum2);

add(num2);

divide=(Button)add(b);

add(rresult);

add(result);

num1.addActionListener(this);

num2.addActionListener(this);

divide.addActionListener(this);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent we)

{

System.exit(0);

}

});

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==divide)

{

try

{

msg=""+Integer.parseInt(num1.getText())/Integer.parseInt(num2.getText());

String c=""+msg;

result.setText(c);

msg1="";

}catch(NumberFormatException e)

{

msg1="Entered number is not an integer "+e;

SampleDialog d = new SampleDialog(this, "Dialog");

d.setVisible(true);

}

catch(ArithmeticException e)

{

msg1="number 2 is zero "+e;

SampleDialog d = new SampleDialog(this, "Dialog");

d.setVisible(true);

}

}

}

public static void main(String[] args)

{

lab12 aa=new lab12();

aa.setSize(new Dimension(400,200));

aa.setTitle("divide\_awt");

aa.setVisible(true);

}

}

Output

