OOJ Lab Record-

Anitej Prasad

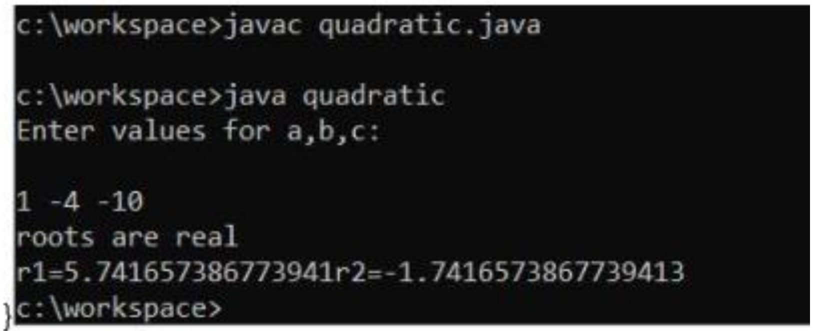
1BM19CS194

3-D

Lab1:

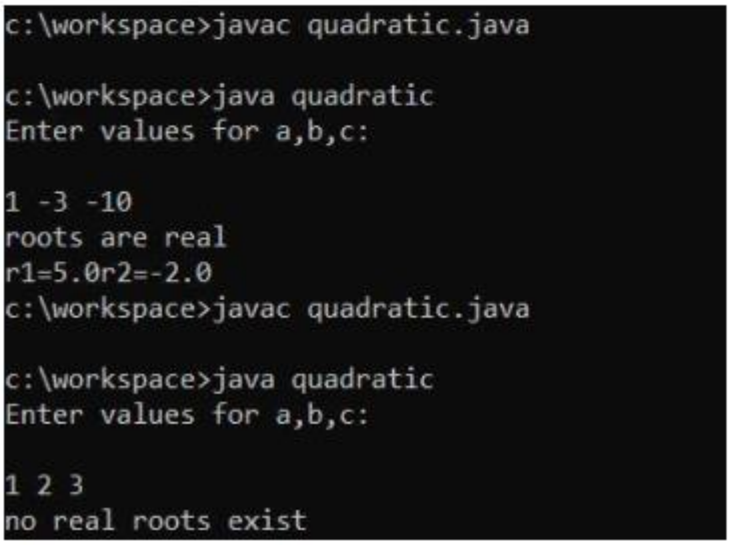
Develop a Java program that prints all real solutions to the quadratic equation ax2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b2-4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
class quadratic {
public static void main(String[] args) {
 double a,b,c,disc;
 double r1, r2;
 Scanner inp= new Scanner(System.in);
 System.out.println("Enter values for a,b,c:\n");
 a=inp.nextDouble();
 b=inp.nextDouble();
 c=inp.nextDouble();
 disc=((b*b)-(4*a*c));
if(disc>0){
 System.out.println("roots are real");
 r1 = (-b+ Math.sqrt(disc))/(2*a);
 r2 = (-b- Math.sqrt(disc))/(2*a);
 System.out.println("r1=" +r1+ "r2="+r2);
}


else if(disc == 0) {
System.out.println("roots are real and equal");
r1=r2 =-b/(2*a);
}


else {
System.out.println("no real roots exist");


}
}
```



```
c:\workspace>javac quadratic.java

c:\workspace>java quadratic
Enter values for a,b,c:

1 -4 -10
roots are real
r1=5.741657386773941r2=-1.7416573867739413
c:\workspace>
```

}

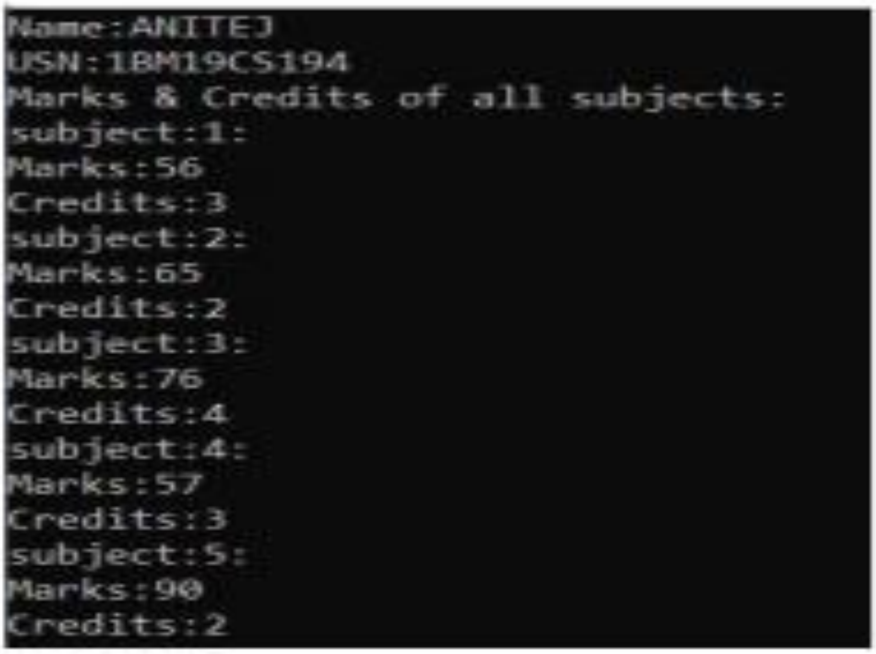


```
c:\workspace>javac quadratic.java

c:\workspace>java quadratic
Enter values for a,b,c:

1 -3 -10
roots are real
r1=5.0r2=-2.0
c:\workspace>javac quadratic.java

c:\workspace>java quadratic
Enter values for a,b,c:

1 2 3
no real roots exist
```

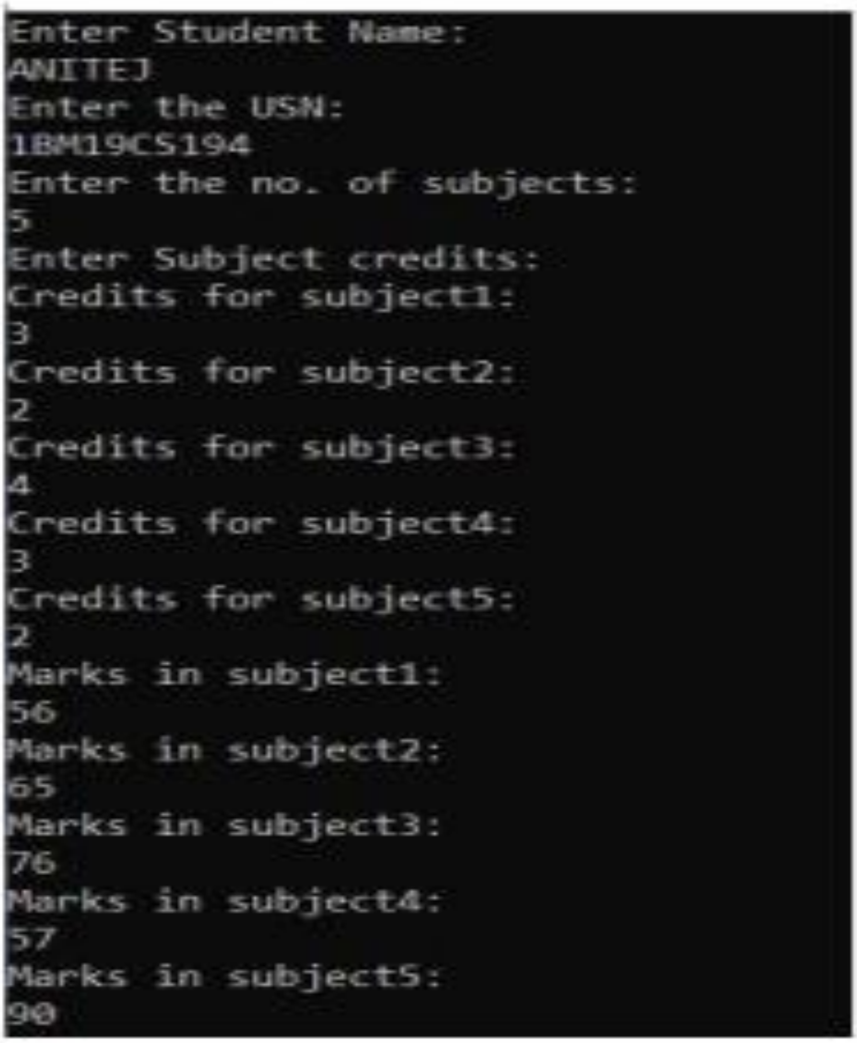
Lab2:

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.

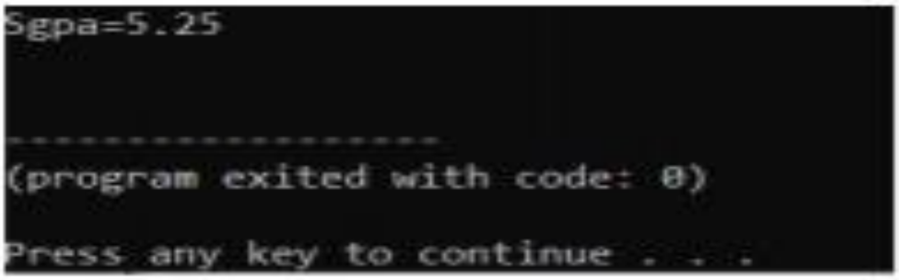
```
import java.util.Scanner;
class Student
{
 String name;
 String usn;
 int marks[] = new int[5];
 int credits[] = new int[5];
 int i,n;
 int grade=0;
 double total=0;
void get_data()
{
 Scanner in = new Scanner(System.in);
 System.out.println("Enter Student Name:");
 name = in.next();
 System.out.println("Enter the USN:");
 usn = in.next();
 System.out.println("Enter the no. of subjects:");
 n=in.nextInt();
 System.out.println("Enter Subject credits:");
 for(i=0;i<n;i++)
 {
 System.out.println("Credits for subject"+(i+1)+":");
 credits[i] = in.nextInt();
}
for(i=0;i<n;i++)
{
 System.out.println("Marks in subject"+(i+1)+":");
 marks[i] = in.nextInt();
 }
}
void calculate_sgpa()
{
 for(i=0;i<n;i++)
 {
 if(marks[i]>=90 && marks[i]<=100)
 grade=10;
 if(marks[i]>=80 && marks[i]<=90)
 grade=9;
 if(marks[i]>=70 && marks[i]<=80)
 grade=8;
 if(marks[i]>=60 && marks[i]<=70)
 grade=7;
 if(marks[i]>=50 && marks[i]<=60)
 grade=6;
 if(marks[i]>=40 && marks[i]<=50)
 grade=5;
 if(marks[i]>=0 && marks[i]<=40)
 grade=0;
 else
 System.out.println("Invalid marks entered");
 total=total+(grade*credits[i]);
 }
 total=total/20;
 System.out.println("Sgpa="+total);
 }
void stud_details()
{
 System.out.println("Name:"+name);
 System.out.println("USN:"+usn);
 System.out.println("Marks & Credits of all subjects:");
 for(i=0;i<n;i++)
 {
 System.out.println("subject:"+(i+1)+":");
 System.out.println("Marks:"+marks[i]);
 System.out.println("Credits:"+credits[i]);
 }
 calculate_sgpa();
}
public static void main(String args[])
{
 Student s = new Student();
 s.get_data();
 s.calculate_sgpa();
 s.stud_details();
}
}
```



```
Name:ANITEJ
USN:1BM19CS194
Marks & Credits of all subjects:
subject:1:
Marks:56
Credits:3
subject:2:
Marks:65
Credits:2
subject:3:
Marks:76
Credits:4
subject:4:
Marks:57
Credits:3
subject:5:
Marks:90
Credits:2
```



```
Enter Student Name:
ANITEJ
Enter the USN:
1BM19CS194
Enter the no. of subjects:
5
Enter Subject credits:
Credits for subject1:
3
Credits for subject2:
2
Credits for subject3:
4
Credits for subject4:
3
Credits for subject5:
2
Marks in subject1:
56
Marks in subject2:
65
Marks in subject3:
76
Marks in subject4:
57
Marks in subject5:
90
```

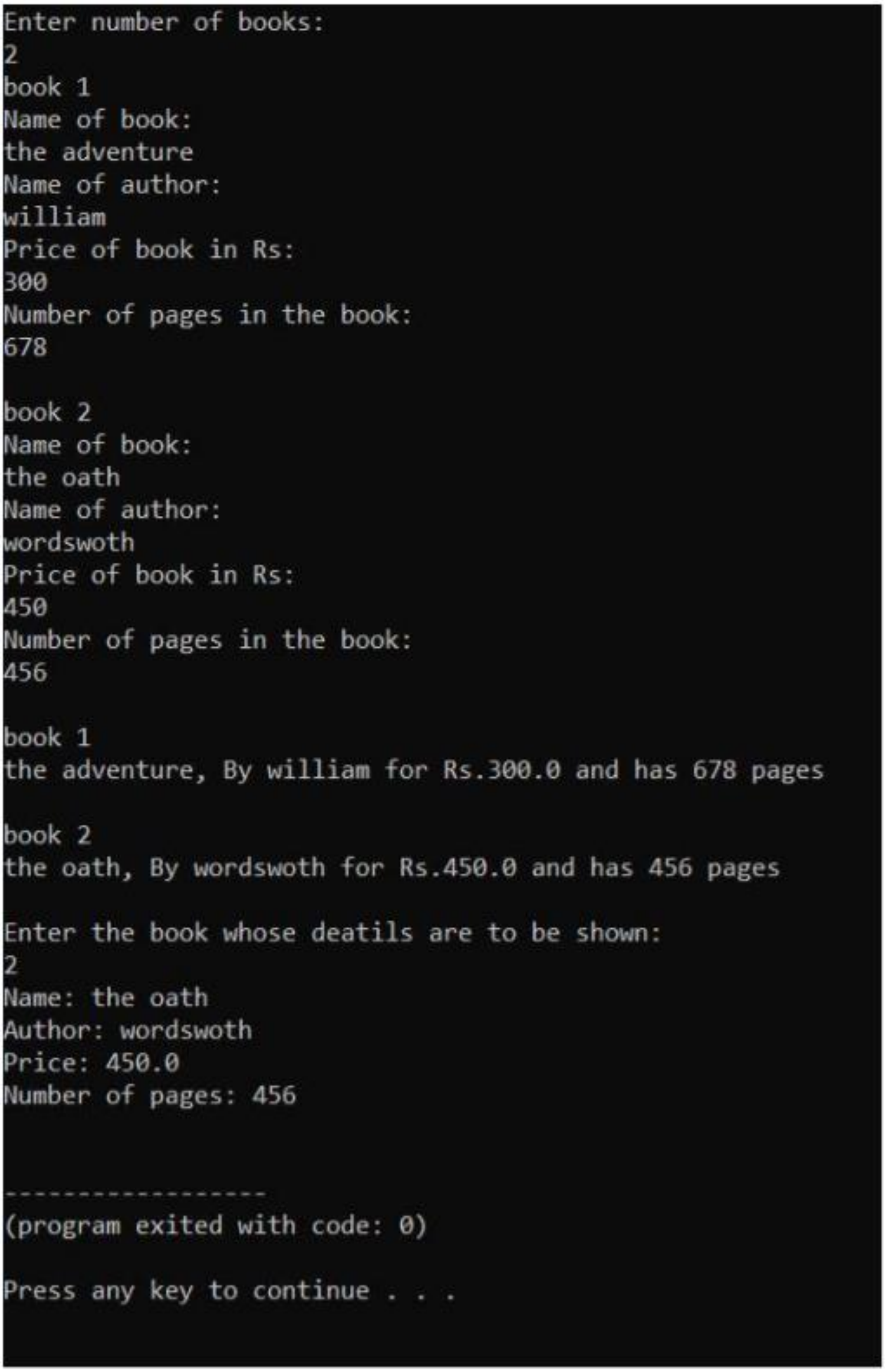


```
Sgpa=5.25


-------------------
(program exited with code: 0)

Press any key to continue . . .
```

Lab3:

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.*;
import java.lang.*; import java.util.*;
import java.lang.*;
class book {
String name, author;
double price;
int num_pages;
Scanner in = new Scanner(System.in);
book() {
System.out.println("Name of book: ");
name = in.nextLine();
System.out.println("Name of author: ");
author = in.nextLine();
System.out.println("Price of book in Rs: ");
price = in.nextDouble();
System.out.println("Number of pages in the book: ");
num_pages = in.nextInt();
}
void display() {
System.out.println("Name: " + name);
System.out.println("Author: " + author);
System.out.println("Price: " + price);
System.out.println("Number of pages: " + num_pages);
}
public String toString() {
return name + ", By " + author + " for Rs." + price + " and has " + num_pages + "
pages";
}
public static void main(String[] args) {
Scanner in = new Scanner(System.in);
int n, x;
System.out.println("Enter number of books: ");
n = in.nextInt();
book B[] = new book[n];
for(int i = 0; i < n; i++) {
System.out.println("book " + (i+1));
B[i] = new book();
System.out.println();
}
for(int i = 0; i < n; i++) {
System.out.println("book " + (i+1));
System.out.println(B[i]);
System.out.println();
}
do {
System.out.println("Enter the book whose deatils are to be shown: ");
x = in.nextInt();
} while(x < 1 && x > n);
B[x-1].display();
}
```



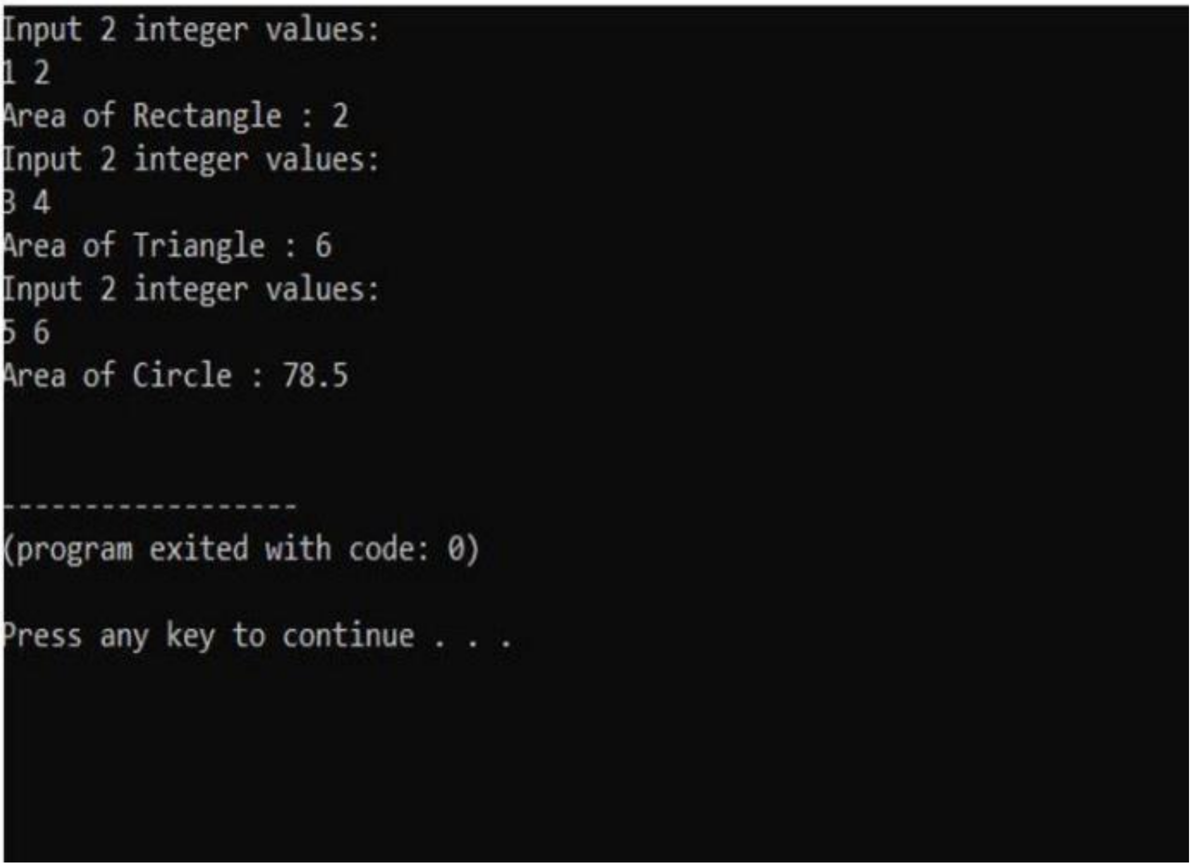
```
Enter number of books:
2
book 1
Name of book:
the adventure
Name of author:
william
Price of book in Rs:
300
Number of pages in the book:
678

book 2
Name of book:
the oath
Name of author:
wordswoth
Price of book in Rs:
450
Number of pages in the book:
456

book 1
the adventure, By william for Rs.300.0 and has 678 pages

book 2
the oath, By wordswoth for Rs.450.0 and has 456 pages

Enter the book whose deatils are to be shown:
2
Name: the oath
Author: wordswoth
Price: 450.0
Number of pages: 456



------------------
(program exited with code: 0)

Press any key to continue . . .
```

Lab4:

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.*;
import java.lang.*;
abstract class Shape {
Scanner in = new Scanner(System.in);
int a1, a2;
Shape() {
System.out.println("Input 2 integer values: ");
a1 = in.nextInt();
a2 = in.nextInt();
}
abstract void printArea();
}
class Rectangle extends Shape {
void printArea() {
System.out.println("Area of Rectangle : " + a1*a2);
}
}
class Triangle extends Shape {
void printArea() {
System.out.println("Area of Triangle : " + (a1*a2)/2);
}
}
class Circle extends Shape {
void printArea() {
```

```
System.out.println("Area of Circle : " + (3.14 * a1 * a1));
}
}
class abstractclassexample {
public static void main(String[] args) {
Shape s;
s = new Rectangle();
s.printArea();
s = new Triangle();
s.printArea();
s = new Circle();
s.printArea();
}
}
```



```
Input 2 integer values:
1 2
Area of Rectangle : 2
Input 2 integer values:
3 4
Area of Triangle : 6
Input 2 integer values:
5 6
Area of Circle : 78.5


-----------------
(program exited with code: 0)

Press any key to continue . . .
```

Lab5:

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.*;
class Account {
String name, abc;
int account_No;
char type_of_account;
double bal = 0;
double deposit;
Scanner in = new Scanner(System.in);
void input_data() {
System.out.println("Enter your account type (Savings/Current):");
abc = in.nextLine();
type_of_account = abc.charAt(0);
}
```

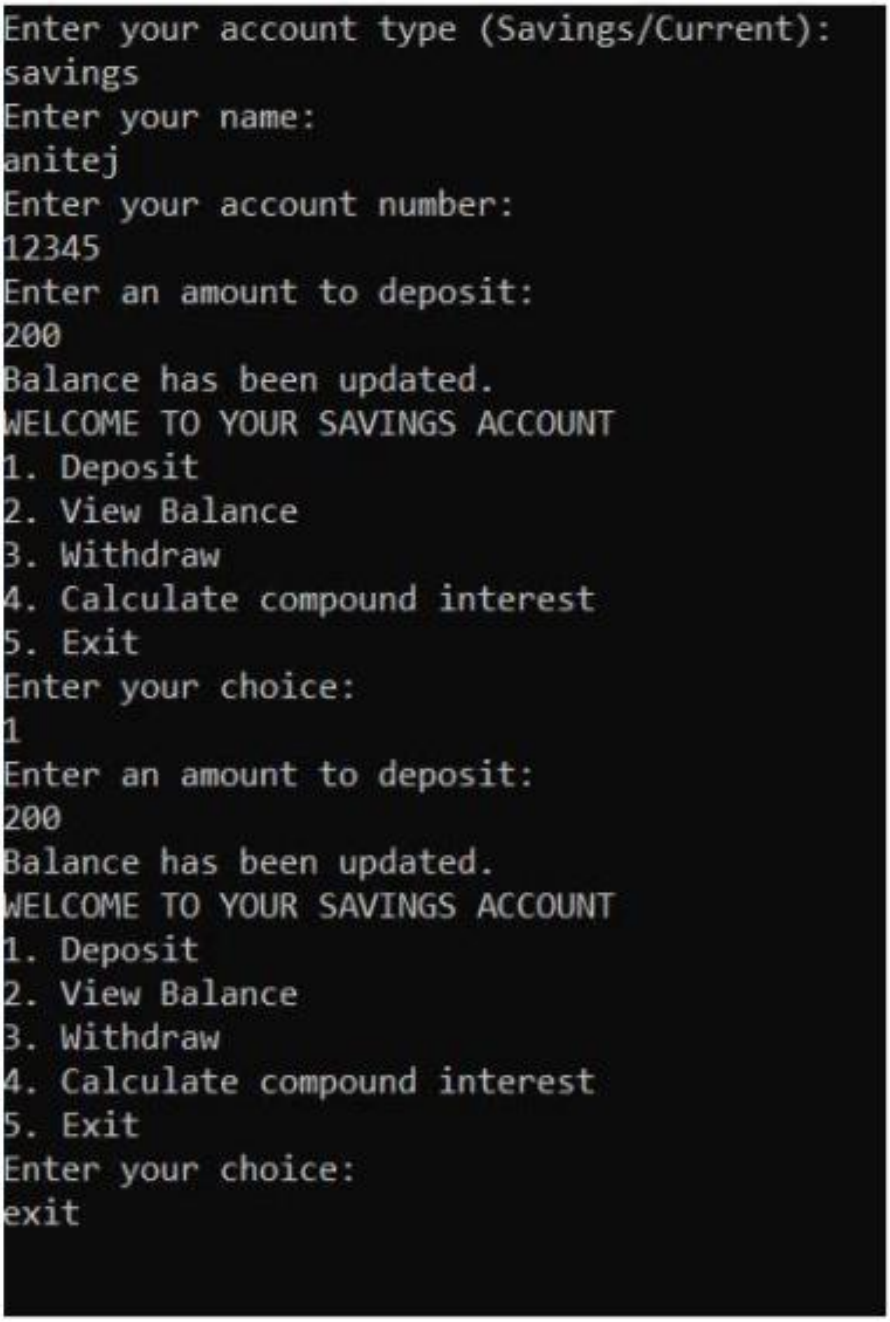
```
void deposit() {
System.out.println("Enter an amount to deposit: ");
deposit = in.nextDouble();
bal += deposit;
System.out.println("Balance has been updated. ");
}
void view_balance() {
System.out.println("Balance = " + bal);
}
public static void main(String[] args) {
Scanner s = new Scanner(System.in);
int x;
Account a1 = new Account();
a1.input_data();
if(a1.type_of_account == 'C' || a1.type_of_account == 'c'){
Current a2 = new Current();
do {
System.out.println("WELCOME TO YOUR CURRENT ACCOUNT");
System.out.println("1. Deposit ");
System.out.println("2. Check Balance ");
System.out.println("3. Issue Cheque ");
System.out.println("4. Exit");
System.out.println("Enter your choice: ");
x = s.nextInt();
switch(x) {
case 1: a2.deposit();
break;
case 2: a2.check_balance();
break;
case 3: a2.issue_cheque();
```

```
break;
case 4: System.exit(0);
break;
default: System.out.println("ERROR. INVALID CHOICE.");
}
} while(x <= 4 && x >= 1);
}
else if (a1.type_of_account == 'S' || a1.type_of_account == 's'){
Savings a3 = new Savings();
do {
System.out.println("WELCOME TO YOUR SAVINGS ACCOUNT");
System.out.println("1. Deposit");
System.out.println("2. View Balance");
System.out.println("3. Withdraw ");
System.out.println("4. Calculate compound interest ");
System.out.println("5. Exit ");
System.out.println("Enter your choice: ");
x = s.nextInt();
switch(x) {
case 1: a3.deposit();
break;
case 2: a3.view_balance();
break;
case 3: a3.withdraw_balance();
break;
case 4: a3.compute_CI();
break;
case 5: System.exit(0);
break;
default: System.out.println("ERROR. INVALID CHOICE.");
```

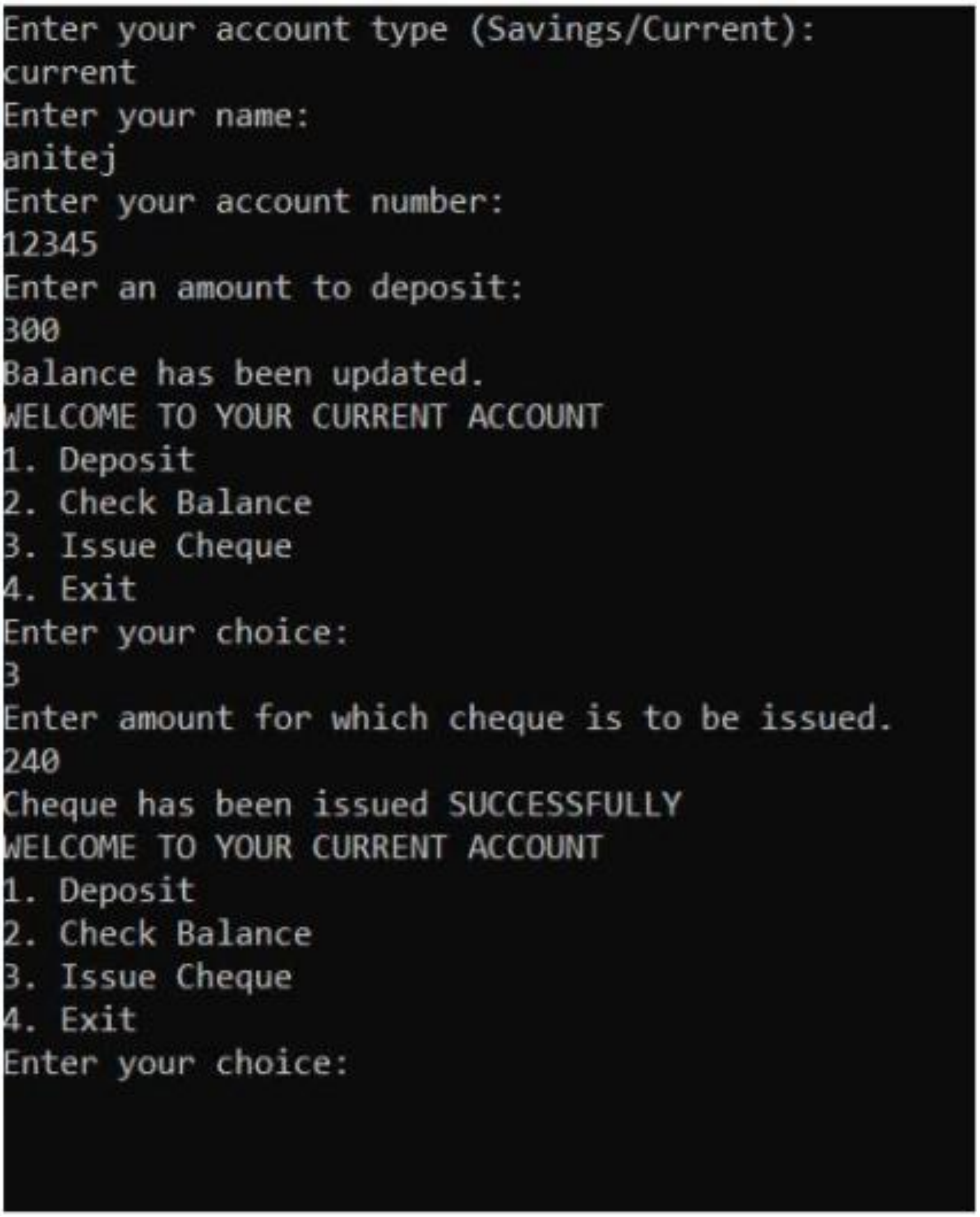
```
}
} while(x <= 5 && x >=1);
}
else System.out.println("INVALID ACCOUNT TYPE");
}
}
class Current extends Account {
Current() {
System.out.println("Enter your name: ");
name = in.nextLine();
System.out.println("Enter your account number: ");
account_No = in.nextInt();
deposit();
}
double chq_amount;
void issue_cheque() {
System.out.println("Enter amount for which cheque is to be issued.");
chq_amount = in.nextDouble();
if(chq_amount > bal) {
System.out.println("ERROR! Insufficient balance in account.");
}
else {
bal -= chq_amount;
System.out.println("Cheque has been issued SUCCESSFULLY");
}
}
void check_balance() {
if(bal < 1000) {
System.out.println("Current available balance is lesser than minimum
required balance.");
```

```
bal -= 100;
System.out.println("Service charge of Rs.100 has been deducted from your
balance.");
}
view_balance();
}
}
class Savings extends Account {
double CI, withdrawal_ammount, time;
Savings() {
System.out.println("Enter your name: ");
name = in.nextLine();
System.out.println("Enter your account number: ");
account_No = in.nextInt();
deposit();
}
void compute_CI() {
System.out.println("Enter time period: ");
time = in.nextInt();
CI = bal * Math.pow(1 + (0.08 / 12), 12 * time) - bal;
System.out.println("CI = " + CI);
bal += CI;
System.out.println("CI has been deposited");
 }
void withdraw_balance() {
System.out.println("Enter the amount you want to withdraw: ");
withdrawal_ammount = in.nextDouble();
if(withdrawal_ammount > bal) {
System.out.println("ERROR! THE ENTERED AMOUNT IS GREATER THAN THE
AVAILABLE BALANCE...");
```

```
}
else {
bal -= withdrawal_ammount;
System.out.println("AMOUNT HAS SUCCESSFULLY BEEN WITHDRAWN!");
}
}
}
```



```
Enter your account type (Savings/Current):
savings
Enter your name:
anitej
Enter your account number:
12345
Enter an amount to deposit:
200
Balance has been updated.
WELCOME TO YOUR SAVINGS ACCOUNT
1. Deposit
2. View Balance
3. Withdraw
4. Calculate compound interest
5. Exit
Enter your choice:
1
Enter an amount to deposit:
200
Balance has been updated.
WELCOME TO YOUR SAVINGS ACCOUNT
1. Deposit
2. View Balance
3. Withdraw
4. Calculate compound interest
5. Exit
Enter your choice:
exit
```



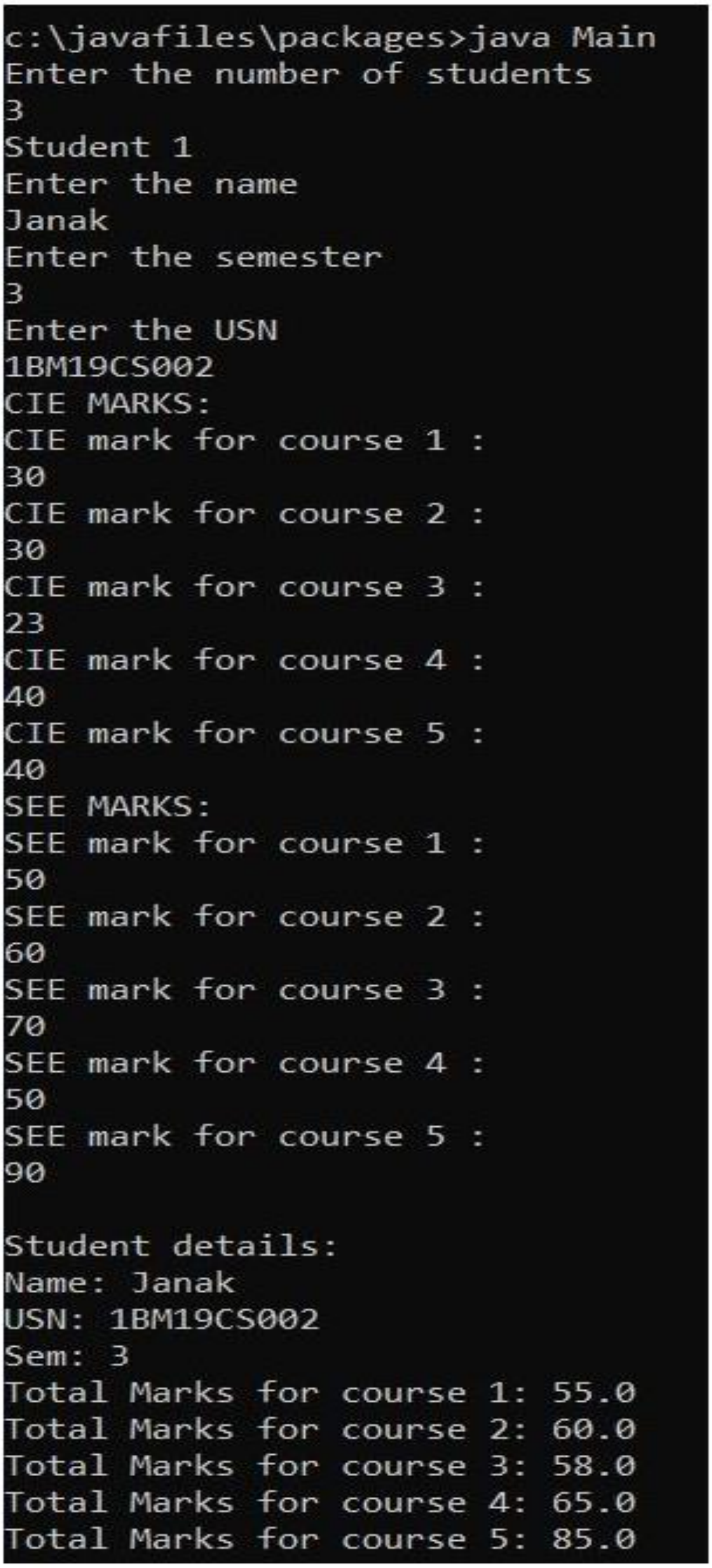
```
Enter your account type (Savings/Current):
current
Enter your name:
anitej
Enter your account number:
12345
Enter an amount to deposit:
300
Balance has been updated.
WELCOME TO YOUR CURRENT ACCOUNT
1. Deposit
2. Check Balance
3. Issue Cheque
4. Exit
Enter your choice:
3
Enter amount for which cheque is to be issued.
240
Cheque has been issued SUCCESSFULLY
WELCOME TO YOUR CURRENT ACCOUNT
1. Deposit
2. Check Balance
3. Issue Cheque
4. Exit
Enter your choice:
```

Lab6:

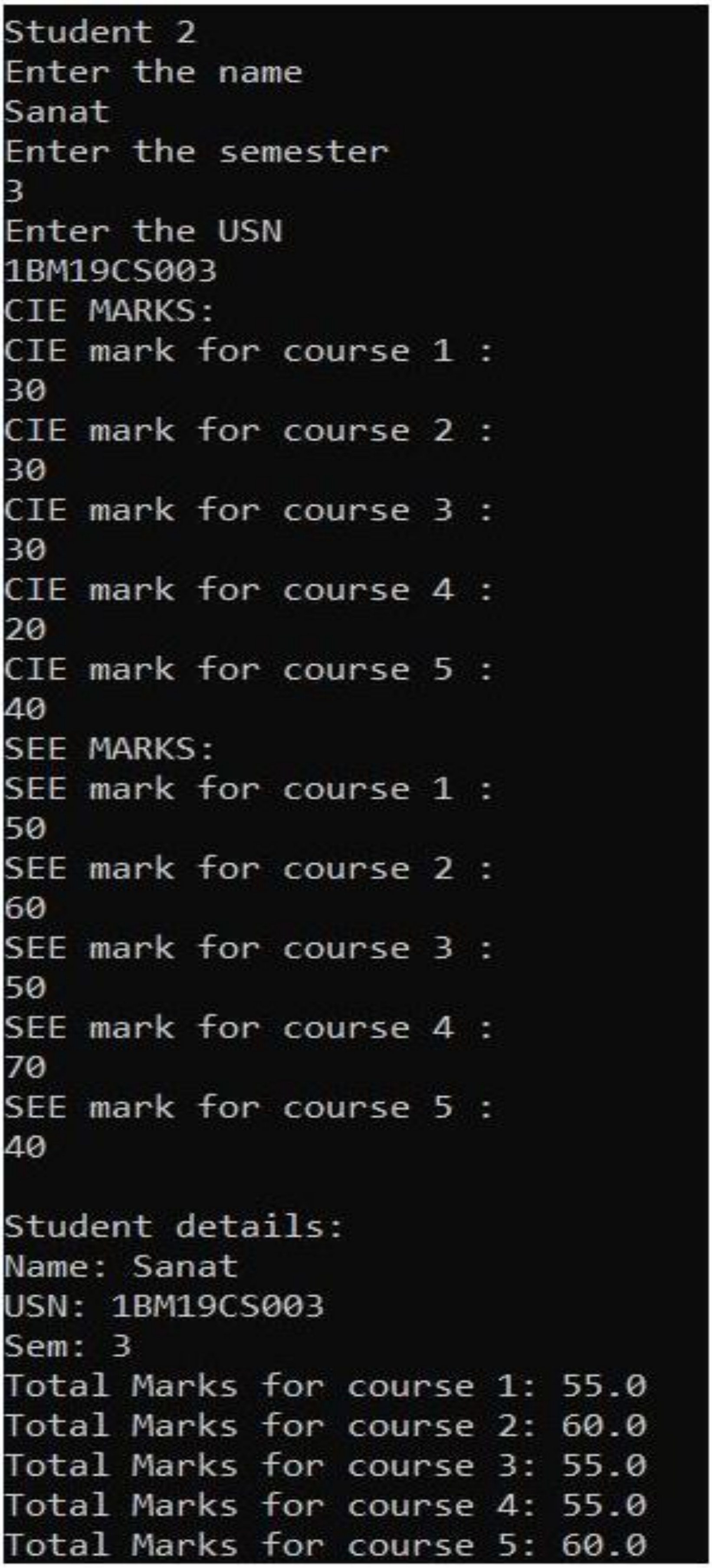
Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External 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 java.util.Scanner;
class Student
{
   String name;
   String usn;
   int marks[] = new int[5];
   int credits[] = new int[5];
   int i,n;
   int grade=0;
   double total=0;

void get_data()
{
   Scanner in = new Scanner(System.in);
   System.out.println("Enter Student Name:");
   name = in.next();
   System.out.println("Enter the USN:");
   usn = in.next();
   System.out.println("Enter the no. of subjects:");
   n=in.nextInt();
   System.out.println("Enter Subject credits and subject marks:");
   for(i=0;i<n;i++)
```

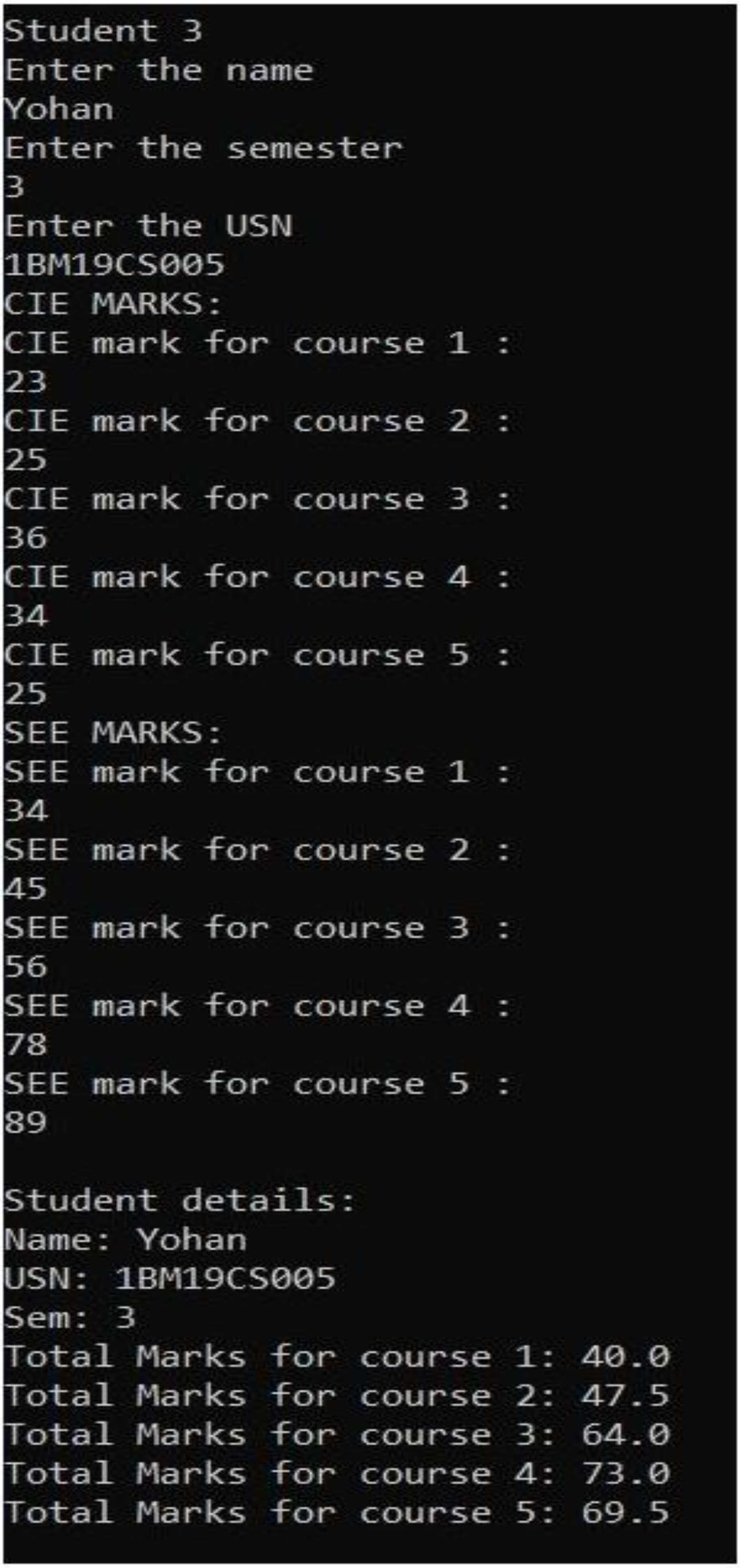
```
    {
        System.out.println("Credits for subject"+(i+1)+":");
        credits[i] = in.nextInt();
        System.out.println("Marks in subject"+(i+1)+":");
        marks[i] = in.nextInt();
    }
}

void calculate_sgpa()
{
    for(i=0;i<n;i++)
    {
        if(marks[i]>=90 && marks[i]<=100)
            grade=10;
        if(marks[i]>=80 && marks[i]<=90)
            grade=9;
        if(marks[i]>=70 && marks[i]<=80)
            grade=8;
        if(marks[i]>=60 && marks[i]<=70)
            grade=7;
        if(marks[i]>=50 && marks[i]<=60)
            grade=6;
        if(marks[i]>=40 && marks[i]<=50)
            grade=5;
        if(marks[i]>=0 && marks[i]<=40)
            grade=0;
        else
        System.out.println("Invalid marks entered");
        total=total+(grade*credits[i]);
    }
    total=total/20;
    System.out.println("Sgpa="+total);
    }

void stud_details()
{
    System.out.println("Name:"+name);
    System.out.println("USN:"+usn);
    System.out.println("Marks & Credits of all subjects:");
    for(i=0;i<n;i++)
    {
        System.out.println("subject:"+(i+1)+":");
        System.out.println("Marks:"+marks[i]);
        System.out.println("Credits:"+credits[i]);
    }
    calculate_sgpa();
}

public static void main(String args[])
{
    Student s = new Student();
    s.get_data();
    s.calculate_sgpa();
    s.stud_details();
}
}
```



```
c:\javafiles\packages>java Main
Enter the number of students
3
Student 1
Enter the name
Janak
Enter the semester
3
Enter the USN
1BM19CS002
CIE MARKS:
CIE mark for course 1 :
30
CIE mark for course 2 :
30
CIE mark for course 3 :
23
CIE mark for course 4 :
40
CIE mark for course 5 :
40
SEE MARKS:
SEE mark for course 1 :
50
SEE mark for course 2 :
60
SEE mark for course 3 :
70
SEE mark for course 4 :
50
SEE mark for course 5 :
90

Student details:
Name: Janak
USN: 1BM19CS002
Sem: 3
Total Marks for course 1: 55.0
Total Marks for course 2: 60.0
Total Marks for course 3: 58.0
Total Marks for course 4: 65.0
Total Marks for course 5: 85.0
```



```
Student 2
Enter the name
Sanat
Enter the semester
3
Enter the USN
1BM19CS003
CIE MARKS:
CIE mark for course 1 :
30
CIE mark for course 2 :
30
CIE mark for course 3 :
30
CIE mark for course 4 :
20
CIE mark for course 5 :
40
SEE MARKS:
SEE mark for course 1 :
50
SEE mark for course 2 :
60
SEE mark for course 3 :
50
SEE mark for course 4 :
70
SEE mark for course 5 :
40

Student details:
Name: Sanat
USN: 1BM19CS003
Sem: 3
Total Marks for course 1: 55.0
Total Marks for course 2: 60.0
Total Marks for course 3: 55.0
Total Marks for course 4: 55.0
Total Marks for course 5: 60.0
```

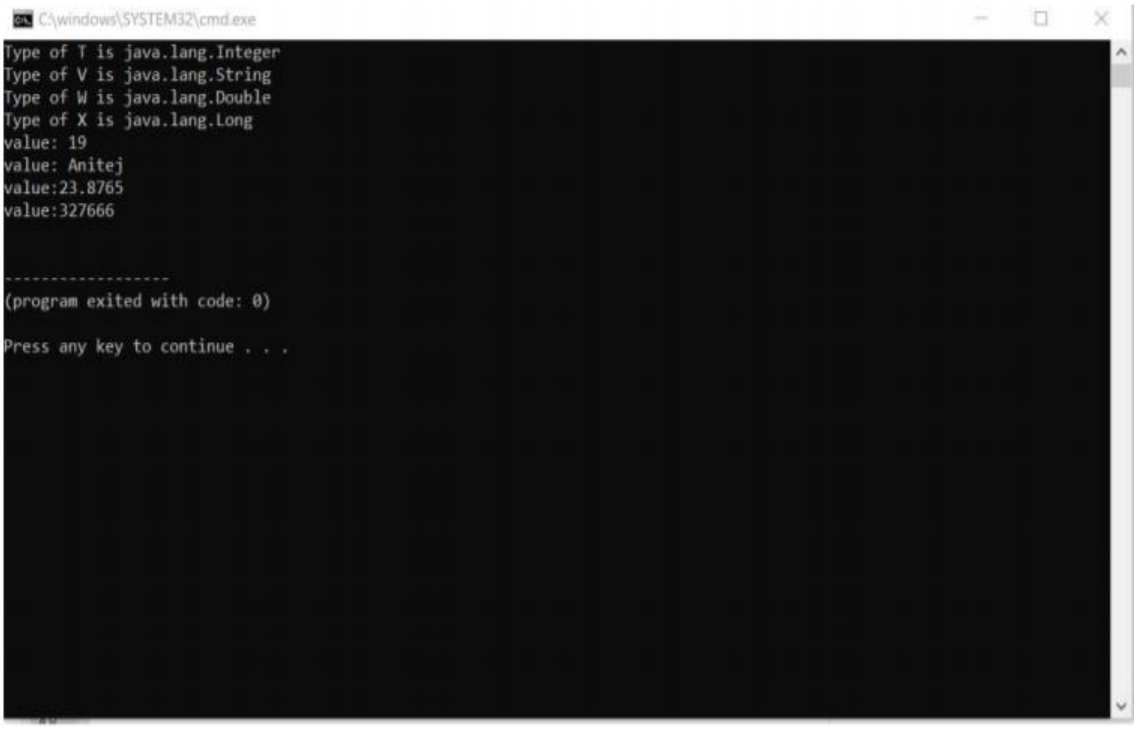


```
Student 3
Enter the name
Yohan
Enter the semester
3
Enter the USN
1BM19CS005
CIE MARKS:
CIE mark for course 1 :
23
CIE mark for course 2 :
25
CIE mark for course 3 :
36
CIE mark for course 4 :
34
CIE mark for course 5 :
25
SEE MARKS:
SEE mark for course 1 :
34
SEE mark for course 2 :
45
SEE mark for course 3 :
56
SEE mark for course 4 :
78
SEE mark for course 5 :
89

Student details:
Name: Yohan
USN: 1BM19CS005
Sem: 3
Total Marks for course 1: 40.0
Total Marks for course 2: 47.5
Total Marks for course 3: 64.0
Total Marks for course 4: 73.0
Total Marks for course 5: 69.5
```

Lab7:

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>(19, "Anitej", 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);

}

}
```



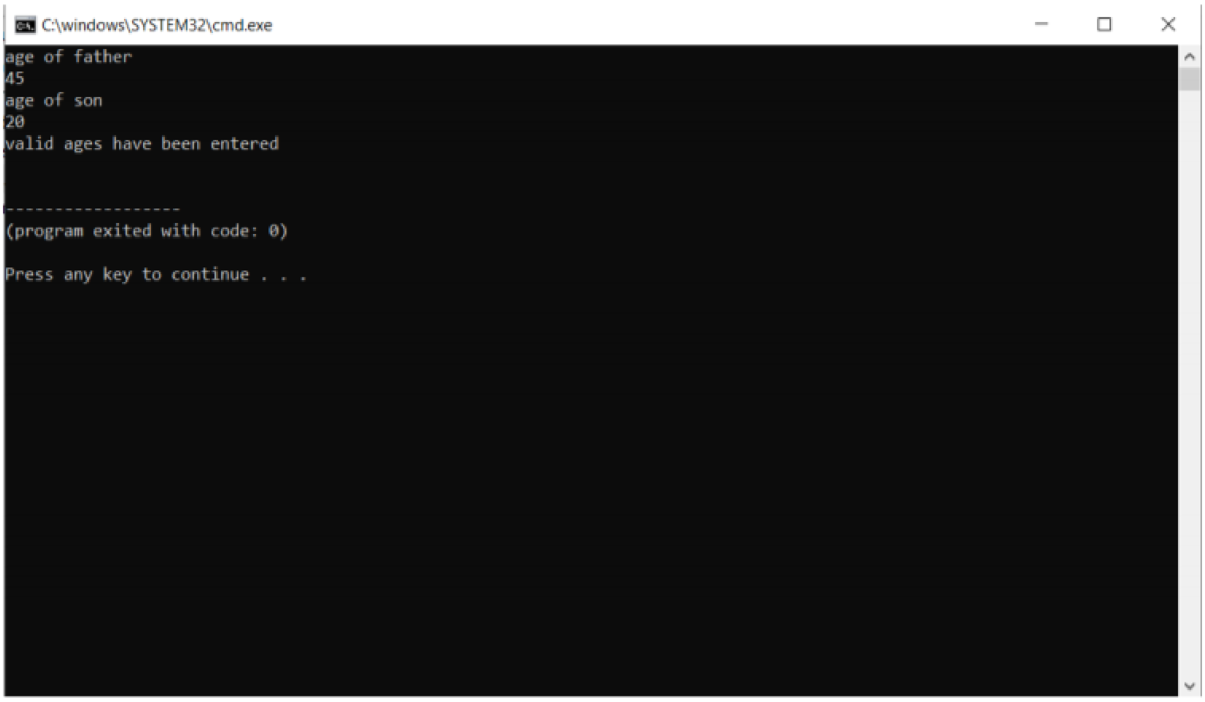
```
Type of T is java.lang.Integer
Type of V is java.lang.String
Type of W is java.lang.Double
Type of X is java.lang.Long
value: 19
value: Anitej
value:23.8765
value:327666


------------------
(program exited with code: 0)

Press any key to continue . . .
```

Lab8:

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.

```
import java.util.*;
class WrongAge extends Exception
{
private int a,b;
WrongAge(int x,int y)
{
a=x;
b=y;
}
public String toString()
{
 if(a<0||b<0)
 return " age cannot be less than 0";
 else if(a<=b)
 return "valid ages have not been entered";
 return "";
}
}
class Father
{
int fage,sage;
Scanner sc=new Scanner(System.in);
```

```
 Father() throws WrongAge
 {
 System.out.println("age of father");
 fage=sc.nextInt();
 System.out.println("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("valid ages have been entered");
 }
}
class FatherSon
{
public static void main(String args[])
{
try
{
 Son s=new Son();
}catch(WrongAge e){
System.out.println("error:"+e);
}
}
}
```



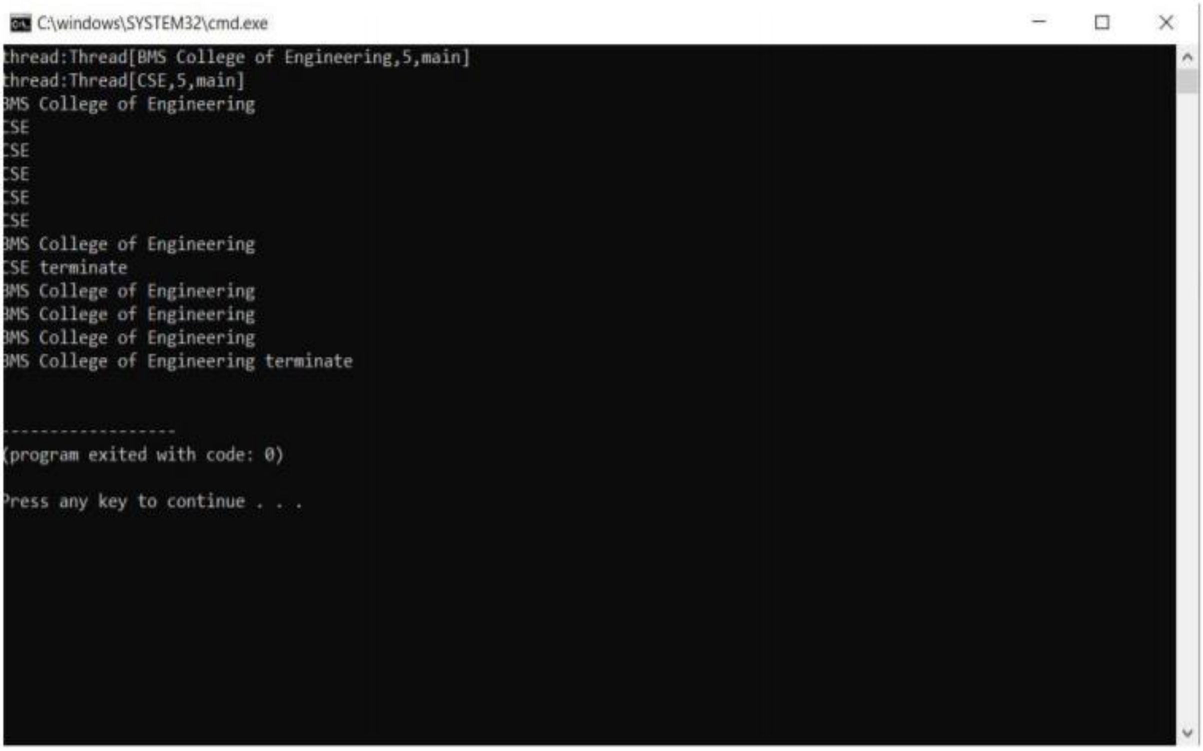
```
age of father
45
age of son
20
valid ages have been entered


------------------
(program exited with code: 0)

Press any key to continue . . .
```

Lab9:

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 thread_test implements Runnable {
 String name;
 Thread t;
 int time;
 thread_test(String Thread_name,int time) {
 name = Thread_name;
 this.time= time;
 t = new Thread(this, name);
 System.out.println("thread:"+ t);
 t.start();
 }
 public void run() {
 try {
 for(int i = 5; i > 0; i--) {
 System.out.println(name);
 Thread.sleep(time);
 }
 } catch (InterruptedException e) {
 System.out.println(name + "Interrupt");
 }
 System.out.println(name + " terminate");
 }
 }
 class threadmain {
 public static void main(String args[]) {
```

```
 thread_test t1= new thread_test("BMS College of Engineering",10000);
 thread_test t2= new thread_test("CSE",2000);


}


}
```



```
thread:Thread[BMS College of Engineering,5,main]
thread:Thread[CSE,5,main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE terminate
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering
BMS College of Engineering terminate


------------------
(program exited with code: 0)

Press any key to continue . . .
```

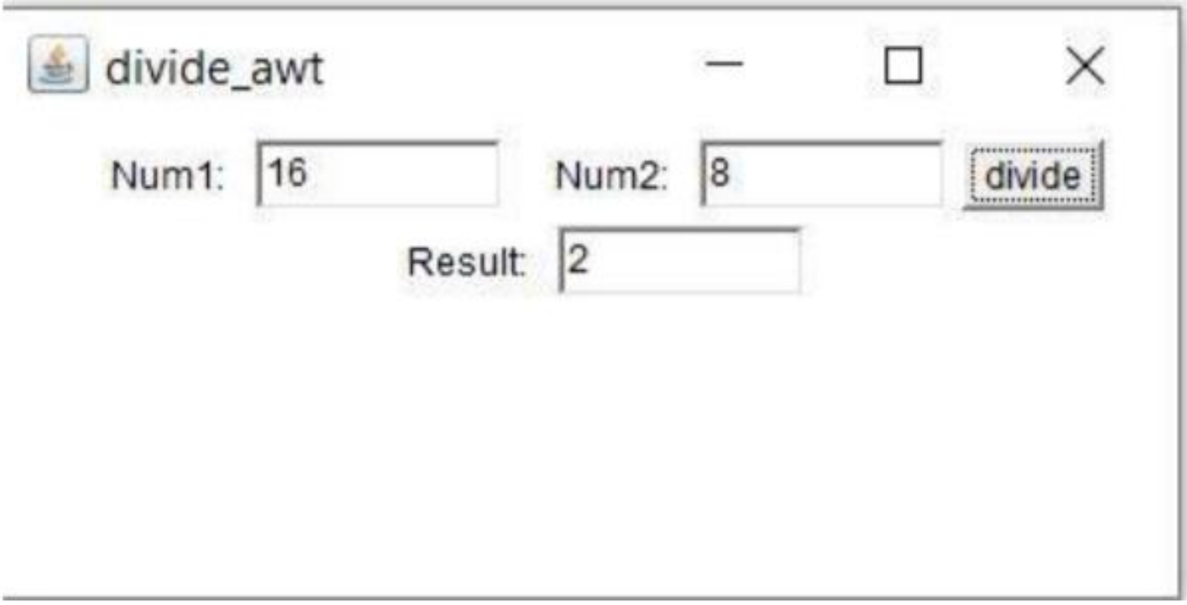
Lab10:

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 {
division bld;
 SampleDialog(Frame parent, String title) {
super(parent, title, false);
 bld=(division)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 division extends Frame implements ActionListener
{
TextField num1,num2,result;
String msg="",msg1="";
```

```
Button divide;
public division()
{
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);
}
}
```



Num1: 16 Num2: 8

Result: 2