

## LAB RECORD : PROGRAMS 1-10

### OOJ LAB

Mallika Prasad

1BM19CS081

#### LAB 1-

Develop a Java program that prints all the real solutions to the quadratic equations  $ax^2+bx+c$ . Read in  $a, b, c$  and use the quadratic formula. If the discriminate  $b^2-4ac$  is negative display a message stating that there are no real solutions.

#### Observation-

CLASSMATE

Date 29.9.2020

Page \_\_\_\_\_

Mallika Prasad Sec-B 1BM19CS081

WEEK 3:  
IP#1

Develop a Java program that prints all real solutions to the quadratic equations  $ax^2+bx+c=0$ . Read in  $a, b, c$  and use the quadratic formula. If the discriminate  $b^2-4ac$  is negative, display a message stating that there are no real solutions.

ALGORITHM:

Step 1 Start

Step 2 Input the value of  $a, b, c$

Step 3 Calculate  $D = (b^2 - 4ac)$

Step 4 If  $(d > 0)$   
    { Display roots are real, calculate the roots  $\Rightarrow x_1 = (-b + \sqrt{D}) / (2 * a)$   
      and  $x_2 = (-b - \sqrt{D}) / (2 * a)$   
    else if  $(d = 0)$   
        Display roots are equal, calculate the roots  $\Rightarrow x_1 = x_2 = -b / (2 * a)$   
    else Display 'There are no real roots'.

Step 5 Print  $x_1$  and  $x_2$

Step 6 Stop

PROGRAM:

```
import java.util.Scanner;  
import java.lang.Math;  
public class Main {  
  
    public static void main (String[] args) {  
        Scanner in = new Scanner(System.in);
```

int a, b, c;

double r1, r2, d;

char ch;

System.out.println("Solution of Quadratic equation  $-ax^2+bx+c$ ");

do

{

System.out.println("\nEnter a: ");

a = in.nextInt();

System.out.println("Enter b: ");

b = in.nextInt();

System.out.println("Enter c: ");

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 -\n" + "r1 = " + r1 + "\n" + "r2 = " + r2);

}

else if (d == 0)

{

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

System.out.println("roots are equal -\n" + "r1 = r2 = " + r1);

}

else

{

System.out.println("there are no real roots");

}

System.out.println("\n" + "do you want to find another set of roots?  
y/n?");

ch = in.next().charAt(0);

}

while (ch == 'y');

{

{

## Output-

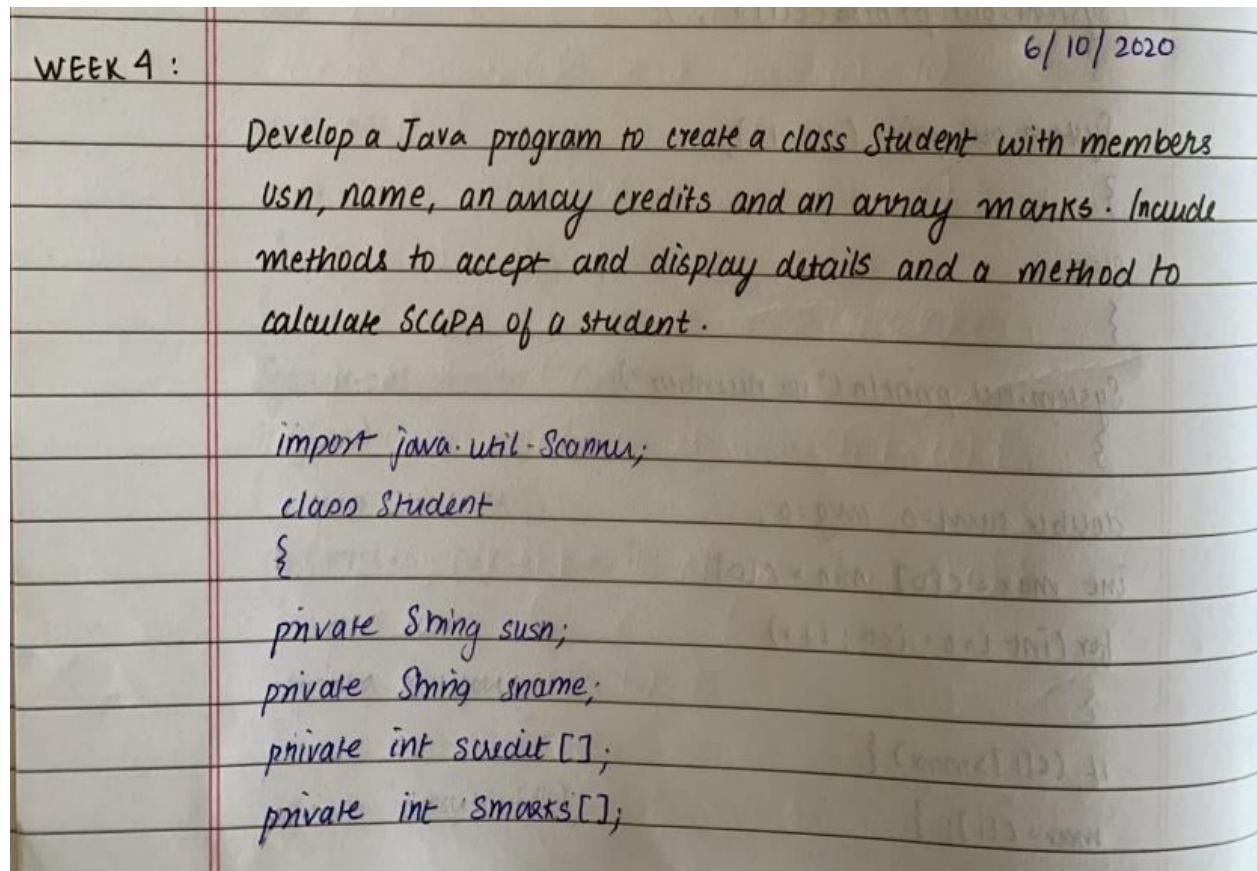
```
input
Solution of Quadratic equation-  $ax^2+bx+c$ 
enter a:
2
enter b:
13
enter c:
4
roots are-
r1= -0.3238250223200936
r2= -6.176174977679906
do you want to find another set of roots? y/n?
y
enter a:
6
enter b:
12
enter c:
6
roots are equal-
r1=r2= -1.0
do you want to find another set of roots? y/n?
y
enter a:
1
```

```
r2= -6.176174977679906
do you want to find another set of roots? y/n?
y
enter a:
6
enter b:
12
enter c:
6
roots are equal-
r1=r2= -1.0
do you want to find another set of roots? y/n?
y
enter a:
1
enter b:
2
enter c:
3
there are no real roots
do you want to find another set of roots? y/n?
n
...Program finished with exit code 0
Press ENTER to exit console.
```

## 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.

### Observation-



WEEK 4 : 6/10/2020

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  
{  
    private String usn;  
    private String name;  
    private int scredit[];  
    private int smarks[];
```



```
void getDetails()  
{  
    System.out.println("enter student details :");  
    Scanner in = new Scanner(System.in);  
    scredit = new int[5];  
    smarks = new int[5];  
    susn = in.next();  
    sname = in.next();  
    for (int i=0; i<5; i++) {  
        System.out.println("credit for sub "+ (i+1) + ":");  
        scredit[i] = in.nextInt();  
    }  
    for (int i=0; i<5; i++) {  
        System.out.println("marks for sub "+ (i+1) + ":");  
        smarks[i] = in.nextInt();  
    }  
}  
  
void printDetails()  
{  
    System.out.println("USN : " + susn);  
    System.out.println("Name : " + sname);  
    for (int i=0; i<5; i++) {  
        System.out.print("credits for sub "+ (i+1) + ":");  
        System.out.println(scredit[i]);  
    }  
}
```

```
for (int i=0; i<5; i++) {  
    System.out.print("marks for sub " + (i+1) + " : ");  
    System.out.println(smarks[i]);  
}
```

```
void sgpa() {  
    int sum=0, sum2=0;  
    double g=0;  
    double sgpa;  
    for (int i=0; i<5; i++) {  
        if (smarks[i] >= 90) {  
            g=10; }  
        else if (smarks[i] >= 80 && smarks[i] < 90)  
            { g=9; }  
        else if (smarks[i] >= 70 && smarks[i] < 80) {  
            g=8; }  
        else if (smarks[i] >= 60 && smarks[i] < 70) {  
            g=7; }  
        else if (smarks[i] >= 50 && smarks[i] < 60) {  
            g=6; }  
        else if (smarks[i] >= 40 && smarks[i] < 50) {  
            g=5; }  
        else if (smarks[i] < 40) {  
            g=0; }  
    }
```

```
sum1 += g * scredit[i];  
sum2 += scredit[i];  
}  
sgpa = sum1 / sum2;  
System.out.print("SGPA of student: ");  
System.out.println(sgpa);  
}  
}
```

```
public class Main  
{
```

```
public static void main (String ss[])  
{
```

```
Student s1 = new Student();
```

```
s1.getDetails();
```

```
s1.printDetails();
```

```
s1.sgpa();
```

```
}
```

```
}
```



### Algorithm :

- Step 1: Start
- Step 2: Input student details i.e. usn, name, credits and marks of each of 5 subjects in 2 different arrays ~~in different arrays~~
- Step 3: Display the student details ~~in array~~
- Step 4: If  $\text{smarks} \geq 90$ ,  $g = 10$   
else if  $\text{smarks} \geq 80$  &  $\text{smarks} < 90$ ,  $g = 9$   
else if  $\text{smarks} \geq 70$  &  $\text{smarks} < 80$ ,  $g = 8$   
else if  $\text{smarks} \geq 60$  &  $\text{smarks} < 70$ ,  $g = 7$   
else if  $\text{smarks} \geq 50$  &  $\text{smarks} < 60$ ,  $g = 6$   
else if  $\text{smarks} \geq 40$  &  $\text{smarks} < 50$ ,  $g = 4$   
else if  $\text{smarks} < 40$ ,  $g = 0$   
Get value of  $g$  and calculate sum of  $(g + \text{credits})$  (sum)
- Step 5: Calculate sgpa Get to sum of credits (sum2)
- Step 6: Calculate  $\text{sgpa} = \frac{\text{sum}}{\text{sum2}}$
- Step 7: Print sgpa of student ~~in array~~
- Step 8: Stop

## Output-

```
enter student details:
ihml9cs081
mal
credit for sub 1:
1
credit for sub 2:
2
credit for sub 3:
3
credit for sub 4:
4
credit for sub 5:
5
marks for sub 1:
10
marks for sub 2:
20
marks for sub 3:
30
marks for sub 4:
40
marks for sub 5:
50
USN: ihml9cs081
Name: mal
credits for sub 1:1
credits for sub 2:2
credits for sub 3:3
```

```
5
marks for sub 1:
10
marks for sub 2:
20
marks for sub 3:
30
marks for sub 4:
40
marks for sub 5:
50
USN: ihml9cs081
Name: mal
credits for sub 1:1
credits for sub 2:2
credits for sub 3:3
credits for sub 4:4
credits for sub 5:5
marks for sub 1:10
marks for sub 2:20
marks for sub 3:30
marks for sub 4:40
marks for sub 5:50
CGPA of student: 3.0

...Program finished with exit code 0
Press ENTER to exit console.
```

```
input
enter student details:
ibml9cs790
sam
credit for sub 1:
4
credit for sub 2:
5
credit for sub 3:
4
credit for sub 4:
6
credit for sub 5:
3
marks for sub 1:
80
marks for sub 2:
70
marks for sub 3:
90
marks for sub 4:
78
marks for sub 5:
55
USN: ibml9cs790
Name: sam
credits for sub 1:4
credits for sub 2:5
credits for sub 3:4
```

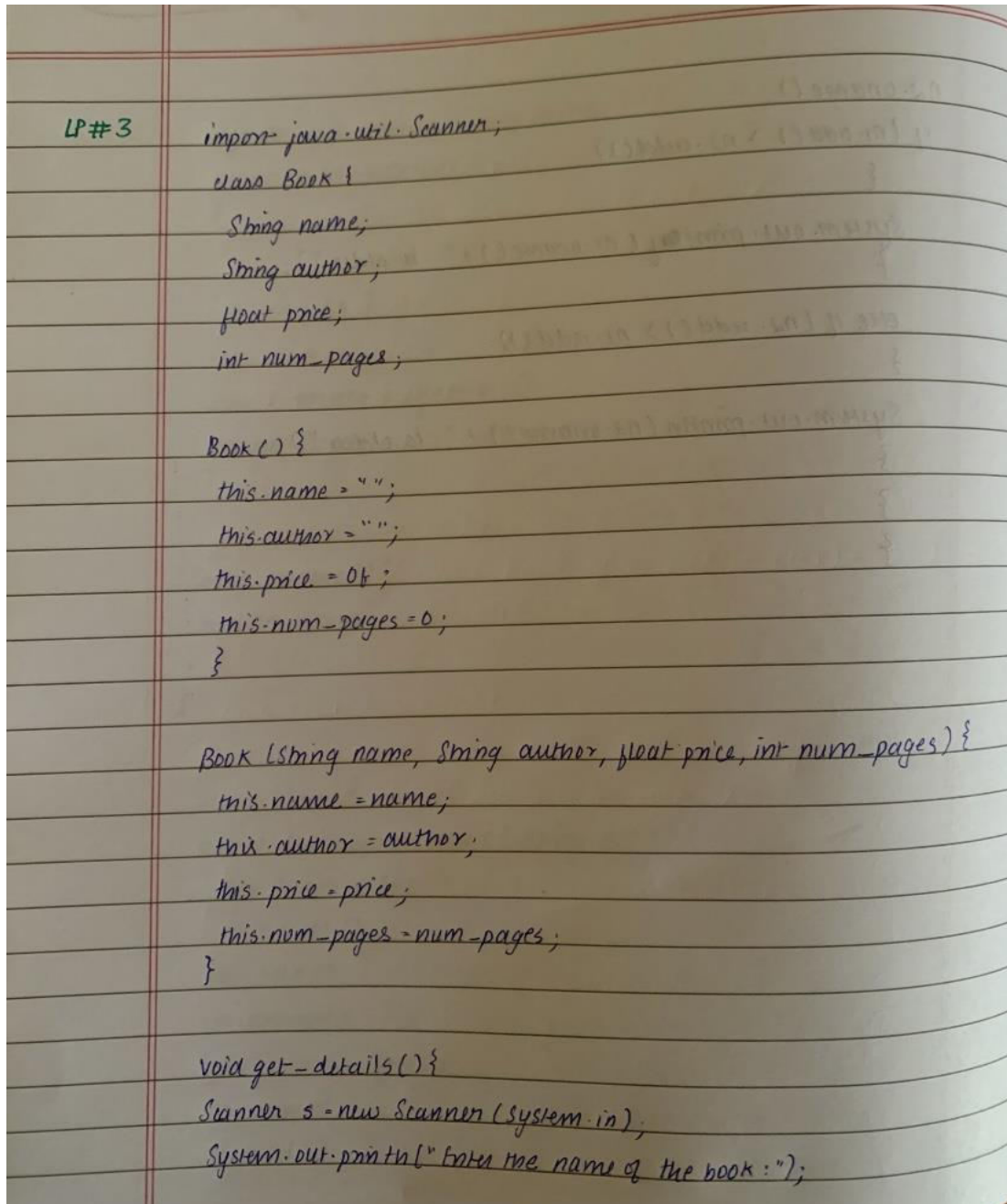
```
3
marks for sub 1:
80
marks for sub 2:
70
marks for sub 3:
90
marks for sub 4:
78
marks for sub 5:
55
USN: ibml9cs790
Name: sam
credits for sub 1:4
credits for sub 2:5
credits for sub 3:4
credits for sub 4:6
credits for sub 5:3
marks for sub 1:80
marks for sub 2:70
marks for sub 3:90
marks for sub 4:78
marks for sub 5:55
SGPA of student: 8.0

...Program finished with exit code 0
Press ENTER to exit console.
```

### 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.

### Observation-



LP#3

```
import java.util.Scanner;

class Book {
    String name;
    String author;
    float price;
    int num_pages;

    Book() {
        this.name = "";
        this.author = "";
        this.price = 0f;
        this.num_pages = 0;
    }

    Book (String name, String author, float price, int num_pages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }

    void get-details() {
        Scanner s = new Scanner (System.in);
        System.out.println("Enter the name of the book :");
```



```
name = s.nextLine();  
System.out.println("Enter the author :");  
author = s.nextLine();  
System.out.println("Enter price of the book :");  
price = s.nextFloat();  
System.out.println("Enter number of pages of the book :");  
num_pages = s.nextInt();  
}
```

```
void set-details (String n, String a, float p, int np) {  
    this.name = n;  
    this.author = a;  
    this.price = p;  
    this.num_pages = np;  
}
```

```
public String toString() {  
    return ("Name = " + name + "\n Author = " + author + "\n Price = " +  
    price + "\n Number of pages = " + num_pages + "\n");  
}
```

```
class Main {  
    public static void main (String args[]) {
```

```
int n;  
Scanner s = new Scanner(System.in);  
System.out.println("Enter number of books:");  
n = s.nextInt();  
Book[] b = new Book[n];  
for (int i = 0; i < n; i++) {  
    b[i] = new Book();  
    b[i].get-details();  
}  
  
for (int i = 0; i < n; i++) {  
    System.out.println("Details of the book " + (i+1) + " :");  
    System.out.println(b[i]);  
}  
  
Book b001 = new Book();  
b001.set-details("The strong world", "Kenny", 300, 400);  
System.out.println("Details of the book:");  
System.out.println(b001);  
}  
}
```

## Output-

```
input
Enter number of books:
2
Enter the name of the book:
harrypotter
Enter the author :
jkr
Enter the price of the book:
333
Enter the number of pages of the book:
234
Enter the name of the book:
divergent
Enter the author :
vr
Enter the price of the book:
444
Enter the number of pages of the book:
312

Details of the book1:
Name=harrypotter
Author=jkr
Price=333.0
Number of pages=234

Details of the book2:
Name=divergent
Author=vr
Price=444.0
Number of pages=312
```

```
divergent
Enter the author :
vr
Enter the price of the book:
444
Enter the number of pages of the book:
312

Details of the book1:
Name=harrypotter
Author=jkr
Price=333.0
Number of pages=234

Details of the book2:
Name=divergent
Author=vr
Price=444.0
Number of pages=312

Details of the book:
Name=The wrld
Author=Kenny
Price=300.0
Number of pages=400

...Program finished with exit code 0
Press ENTER to exit console.
```

## 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.

### Observation-

classmate

Date 3/11/2020

Page

WEEK 8  
LP#4:

Develop a java program to create an abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide 3 classes named Rectangle, Triangle, and Circle such that each one of these classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
abstract class Shape {  
    int dim1;  
    int dim2;  
    Shape(int a, int b)  
    {  
        dim1 = a;  
        dim2 = b;  
    }  
  
    Shape(int a)  
    { dim1 = a; }  
    abstract double PrintArea();  
}  
  
class Rectangle extends Shape {  
    Rectangle(int a, int b)  
    { super(a, b); }  
    double PrintArea() {  
        System.out.println("Inside area of Rectangle :");  
        return dim1 * dim2;  
    }  
}
```

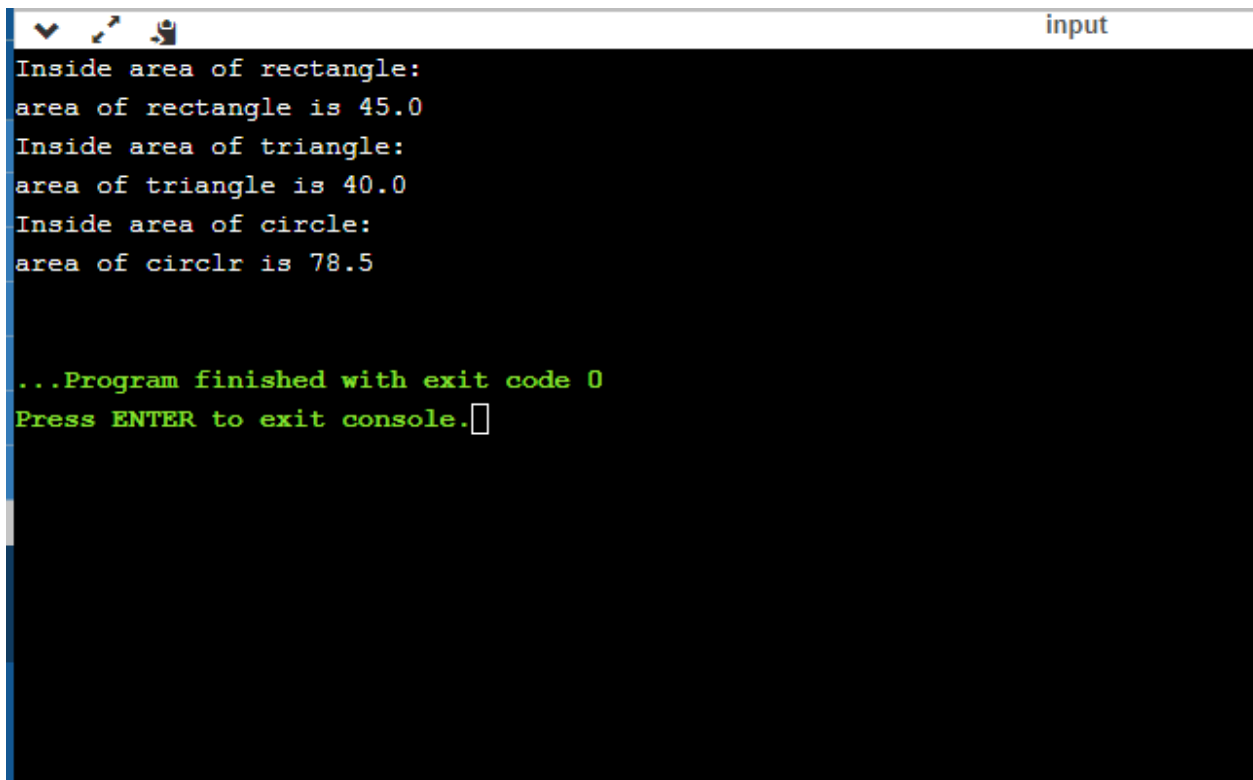


```
}  
}  
class Triangle extends Shape {  
    Triangle (int a, int b) { super (a, b); }  
    double PrintArea () {  
        System.out.println ("Inside area of triangle : ");  
        return dim1 * dim2 / 2;  
    }  
}
```

```
class Circle extends Shape {  
    Circle (int a) { super (a); }  
    double PrintArea () {  
        System.out.println ("Inside area of circle : ");  
        return 3.14 * dim1 * dim1;  
    }  
}
```

```
class Main {  
    public static void main (String ss[]) {  
        Rectangle r = new Rectangle (9, 5);  
        Triangle t = new Triangle (10, 8);  
        Circle c = new Circle (5);  
        System.out.println ("area of rectangle is " + r.PrintArea());  
        System.out.println ("area of triangle is " + t.PrintArea());  
        System.out.println ("area of circle is " + c.PrintArea());  
    }  
}
```

## Output-

A screenshot of a Java IDE's console window. The window has a title bar with standard OS icons and the word "input" on the right. The console background is black with white text. The output shows the program calculating the areas of a rectangle, triangle, and circle. The rectangle area is 45.0, the triangle area is 40.0, and the circle area is 78.5. The program then finishes with exit code 0 and prompts the user to press ENTER to exit the console.

```
Inside area of rectangle:  
area of rectangle is 45.0  
Inside area of triangle:  
area of triangle is 40.0  
Inside area of circle:  
area of circle is 78.5  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

## 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:*

- Accept deposit from customer and update the balance.
- Display the balance.
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for the minimum balance, impose penalty if necessary and update the balance

## Observation-

classmate

Date 11/3/2020

Page

LP #5:

Develop a program to create a class Bank that maintains 2 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 facilities. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum of balance and if the balance falls below this level, a service charge is imposed.

Create a class Account that stores customer name, account number & type of account. From this derive the classes Cur-acc and Sav-acc to make them more specific to their requirements.

Include the necessary methods in order to achieve the following tasks

- Accept deposit from customer & update the balance.

- Display the balance
- Compute and deposit interest
- Permit withdrawal and update the balance
- Check for minimum balance, impose penalty if necessary and update the balance

```
import java.util.Scanner
```

```
class Account {
```

```
    String name, type;
```

```
    int accno;
```

```
    float bal = 0; float pen = 100;
```

```
    Account () {
```

```
        Scanner ss = new Scanner(System.in);
```



```
System.out.println("Enter name, account type number and type  
of account:");
```

```
name = ss.next();
```

```
accno = ss.nextInt();
```

```
type = ss.next();
```

```
}
```

```
}
```

```
class cur acc extends Account {
```

```
float dep, wit;
```

```
void deposit() {
```

```
Scanner ss = new Scanner(System.in);
```

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

```
dep = ss.nextFloat();
```

```
bal = bal + dep;
```

```
System.out.println("Updated balance after deposit: " + bal);
```

```
}
```

```
void withdraw() {
```

```
Scanner ss = new Scanner(System.in);
```

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

```
wit = ss.nextFloat();
```

```
bal = bal - wit;
```

```
System.out.println("Updated balance after withdrawal: " + bal);
```



}

```
void penalty() {
```

```
if (bal < 500)
```

{

```
    bal = bal - pen;
```

```
    System.out.println("Updated balance after imposing penalty : "+bal);
```

}

```
else
```

```
    System.out.println("No penalty imposed, balance : "+bal);
```

}

}

```
class sav acct extends Account {
```

```
    float dep, wit, r, t, n, ci;
```

```
    void deposit() {
```

```
        Scanner ss = new Scanner(System.in);
```

```
        System.out.println("enter amount to be deposited : ");
```

```
        dep = ss.nextFloat();
```

```
        System.out.println("enter rate %, time in years and number  
of times interest is compounded per year : ");
```

```
        r = ss.nextFloat();
```

```
        t = ss.nextFloat();
```

```
        n = ss.nextInt();
```

```

ci = dep * ((float) Math.pow ((1 + (r/100 + n)), (n + t)));
bal = bal + ci;
System.out.println ("Updated balance after computing ci: " + bal);
}

```

```

void withdraw () {
    Scanner ss = new Scanner (System.in);
    System.out.println ("enter amount to be withdrawn:");
    wit = ss.nextInt();
    bal = bal - wit;
    System.out.println ("Updated balance after withdrawal: " + bal);
}
}

```

```

class Bank {
    public static void main (String sss []) {
        Scanner ss = new Scanner (System.in);
        int opt; int ch;
        System.out.println ("choose type of account:");
        System.out.println ("1. savings account\n2. current account");
        opt = ss.nextInt();
        if (opt == 1) {
            System.out.println ("****SAVINGS ACCOUNT****");
            System.out.println ("_____ no chequebook facilities available_____");
            sav_acc s = new sav_acc();

```

```
System.out.println("1. deposit with compound interest\n 2. with-  
draw\n 3. exit\n");
```

```
do
```

```
{
```

```
System.out.println("enter choice ");
```

```
ch = ss.nextInt();
```

```
switch (ch)
```

```
{
```

```
case 1 : s.deposit(); break
```

```
case 2 : s.withdraw(); break;
```

```
case 3: break;
```

```
}
```

```
} while (ch != 3);
```

```
}
```

```
if (opt == 2) {
```

```
System.out.println("**** CURRENT ACCOUNT ****");
```

```
System.out.println("_____chequebook services available _____");
```

```
cur_acct c = new cur_acct();
```

```
System.out.println("1. deposit\n 2. withdraw\n 3. check minim-  
um balance / penalty\n 4. exit\n");
```

```
do
```

```
{
```

```
System.out.println("enter choice ");
```

```
ch = ss.nextInt();
```



```
switch (ch)
```

```
{
```

```
case 1 : c.deposit(); break;
```

```
case 2 : c.withdraw(); break;
```

```
case 3 : c.penalty(); break;
```

```
}
```

```
{ while ch != 4;
```

```
{
```

```
}
```

```
}
```



## Output-

### Savings:

```
choose type of account:
1.savings account
2.current account

1
****SAVINGS ACCOUNT****
_____no chequebook services available_____
Enter name, account no. and type of account:
mal 123 savings
1.deposit with compound interest
2.withdraw
3.exit

enter choice
1
enter amount to be deposited:
5000
enter rate%, time in years and number of times interest is compounded per year:
5 10 12
Updated balance after computing CI: 8235.103
enter choice
2
enter amount to be withdrawn:
7000
Updated balance after withdrawal: 1235.1025
enter choice
1
enter amount to be deposited:
100
enter rate%, time in years and number of times interest is compounded per year:
4 8 11
```

```
Enter name, account no. and type of account:
mal 123 savings
1.deposit with compound interest
2.withdraw
3.exit

enter choice
1
enter amount to be deposited:
5000
enter rate%, time in years and number of times interest is compounded per year:
5 10 12
Updated balance after computing CI: 8235.103
enter choice
2
enter amount to be withdrawn:
7000
Updated balance after withdrawal: 1235.1025
enter choice
1
enter amount to be deposited:
100
enter rate%, time in years and number of times interest is compounded per year:
4 8 11
Updated balance after computing CI: 1372.7354
enter choice
3

...Program finished with exit code 0
Press ENTER to exit console.
```

## Current:

```
choose type of account:
1.savings account
2.current account

2
****CURRENT ACCOUNT****
      chequebook services available
Enter name, account no. and type of account:
mal 1234 current
1.deposit
2.withdraw
3.check minimum balance/penalty
4.exit

enter choice
1
Enter amount to be deposited:
2000
updated balance after deposit: 2000.0
enter choice
2
enter amount to be withdrawn:
1400
Updated balance after withdrawal: 600.0
enter choice
3
No penalty imposed, balance:600.0
enter choice
2
enter amount to be withdrawn:
200
```

```
Updated balance after withdrawal: 600.0
enter choice
3
No penalty imposed, balance:600.0
enter choice
2
enter amount to be withdrawn:
200
Updated balance after withdrawal: 400.0
enter choice
3
Updated balace after imposing penalty: 300.0
enter choice
1
Enter amount to be deposited:
500
updated balance after deposit: 800.0
enter choice
2
enter amount to be withdrawn:
100
Updated balance after withdrawal: 700.0
enter choice
3
No penalty imposed, balance:700.0
enter choice
4

...Program finished with exit code 0
Press ENTER to exit console.
```

## 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.

### Observation-

WEEK 9 : 17/11/2020

LP#-6: Create a package CIE which has 2 classes - Student and Internals. The class Student has members like usn, name, sem - The class Internals has an array that stores the internal marks scored in 5 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 5 courses of the current semester of the student. Import the 2 packages in a file that declares the final marks of n students in all 5 courses.

(PACKAGE CIE -)

( student.java )

```
package CIE;
```

```
public class student {
```

```
    public int usn, sem;
```

```
public String name;  
public void get (int u, int s, String n) {  
    uon = u;  
    sem = s;  
    name = n; }  
public void set ()  
{  
    System.out.println ("Name = " + name + " USN = " + uon + " SEM =  
    " + sem); }  
}
```

(internals.java)

```
package CIE;  
import java.util.Scanner;  
public class internals extends CIE.Student {  
    public float inm[];  
    public internals ()  
    {  
        Scanner ss = new Scanner (System.in);  
        inm = new float [5];  
        System.out.println ("internals marks for 5 subjects (out of 50):");  
        for (int i = 0; i < 5; i++) {  
            System.out.println ("subject " + (i+1));  
            inm[i] = ss.nextFloat(); }  
    }
```



```
}
```

```
}
```

(Package SEE -)

(external.java)

```
package SEE;
```

```
import CIE.*;
```

```
import java.util.Scanner;
```

```
public class external extends CIE.Student {
```

```
    public float exm[];
```

```
    public external() {
```

```
        Scanner ss = new Scanner(System.in);
```

```
        exm = new float[5];
```

```
        System.out.println("external marks for 5 subject (out of 100):");
```

```
        for (int i = 0; i < 5; i++) {
```

```
            System.out.println("subject " + (i+1));
```

```
            exm[i] = ss.nextFloat(); }  
    }
```

```
}
```

```
}
```

(Driver class -)

(finalmarks.java)

```
import CIE.*;
```

```
import SEE.*;
```

```

import java.util.Scanner;
class finalmarks {
    public static void main (String args []) {
        Scanner ss = new Scanner (System.in);
        System.out.println ("enter the number of students");
        int no = ss.nextInt();
        for (int j=0 ; j<no; j++) {
            float tot[] = new float [5];
            CIE-student stud = new CIE-student();
            System.out.println ("Enter name, uo and sem");
            String n = ss.next();
            int u = ss.nextInt();
            int s = ss.nextInt();
            stud.set(u, s, n);
            CIE-internals ci = new CIE-internals();
            SEE-external se = new SEE-external();
            stud.set();
            for (int i=0; i<5; i++) {
                tot[i] = (ci.inm[i] + (se.exm[i]/2));
                System.out.println ("Total marks for subject: " + (i+1));
                System.out.println (tot[i]);
            }
        }
    }
}

```

## Output-

```
Command Prompt
enter name, usn and sem
mal 111 3
internals marks for 5 subjects (out of 50):
subject 1
41
subject 2
21
subject 3
35
subject 4
45
subject 5
28
external marks for 5 subjects (out of 100):
subject 1
98
subject 2
65
subject 3
84
subject 4
76
subject 5
54

NAME= mal
USN= 111
SEM= 3
Total marks for subject: 1
90.0
Total marks for subject: 2
53.5
Total marks for subject: 3
77.0
Total marks for subject: 4
83.0
Total marks for subject: 5
55.0
```

```
Command Prompt
enter name, usn and sem
ren 113 4
internals marks for 5 subjects (out of 50):
subject 1
50
subject 2
45
subject 3
39
subject 4
48
subject 5
36
external marks for 5 subjects (out of 100):
subject 1
78
subject 2
99
subject 3
63
subject 4
52
subject 5
100

NAME= ren
USN= 113
SEM= 4
Total marks for subject: 1
89.0
Total marks for subject: 2
94.5
Total marks for subject: 3
70.5
Total marks for subject: 4
74.0
Total marks for subject: 5
86.0
```

- LAB6
  - CIE
    - internals.class
    - student.class
  - SEE
    - external.class
    - external.java
    - finalmarks.class
    - finalmarks.java**
    - internals.java
    - student.java



## LAB 7-

Write a program to demonstrate generics with multiple object parameters.

WEEK 10: 24/11/2020 Mallika Phusad IBM19CS081

LP #7: WAP to demonstrate generics with multiple object parameters

```
class TwoGen<T, V> {
    T obj; V ob2;
    TwoGen (T o1, V o2) {
        obj = o1; ob2 = o2;
    }
    void showTypes () {
        System.out.println ("Type of T is " + obj.getClass().getName());
        System.out.println ("Type of V is " + ob2.getClass().getName());
    }
    T getobj1 () { return obj; }
}
```

```
    V getobj2 () { return ob2; }
}

class SimpleMain {
    public static void main (String args[]) {
        TwoGen<Integer, String> tgObj = new TwoGen<Integer, String>
            (88, "Generics");
        tgObj.showTypes();
        int v = tgObj.getobj1();
        System.out.println ("value : " + v);
        String str = tgObj.getobj2();
        System.out.println ("value: " + str);
    }
}
```

## Output-

```
input
Type of T is java.lang.Integer
Type of V is java.lang.String
value: 88
value: Generics

...Program finished with exit code 0
Press ENTER to exit console.
```

## 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 Wrong Age( when the input age=father's age).

LP#8: WAP 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.

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge extends Exception {
        public WrongAge(String s) {
            super(s);
        }
    }
}
```

```
class Father {  
    int fatherAge;
```

```
    Father(int fAge, int sAge) throws WrongAge {  
        if (fAge <= sAge) { throw new WrongAge("Son's age  
            can't be greater than or equal to the father's age"); }  
        else {  
            this.fatherAge = fAge; }  
        }  
    }
```

```
class Son extends Father {
```

```
    int sonAge;
```

```
    Son(int fAge, int sAge) throws WrongAge {  
        super(fAge, sAge);  
        this.sonAge = sAge;  
    }
```

```
    void print() {
```

```
        System.out.println("Father's age: " + fatherAge);
```

```
        System.out.println("Son's age: " + sonAge);
```

```
    }
```

```
}
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        int fAge, sAge;
```

```
Scanner sc = new Scanner(System.in);  
System.out.println("Enter father's age");  
fAge = sc.nextInt();  
System.out.println("Enter son's age");  
sAge = sc.nextInt();  
try {  
    Son son = new Son(fAge, sAge);  
    son.print();  
} catch (WrongAge err) {  
    System.out.println("Exception ~ " + err);  
}  
}
```



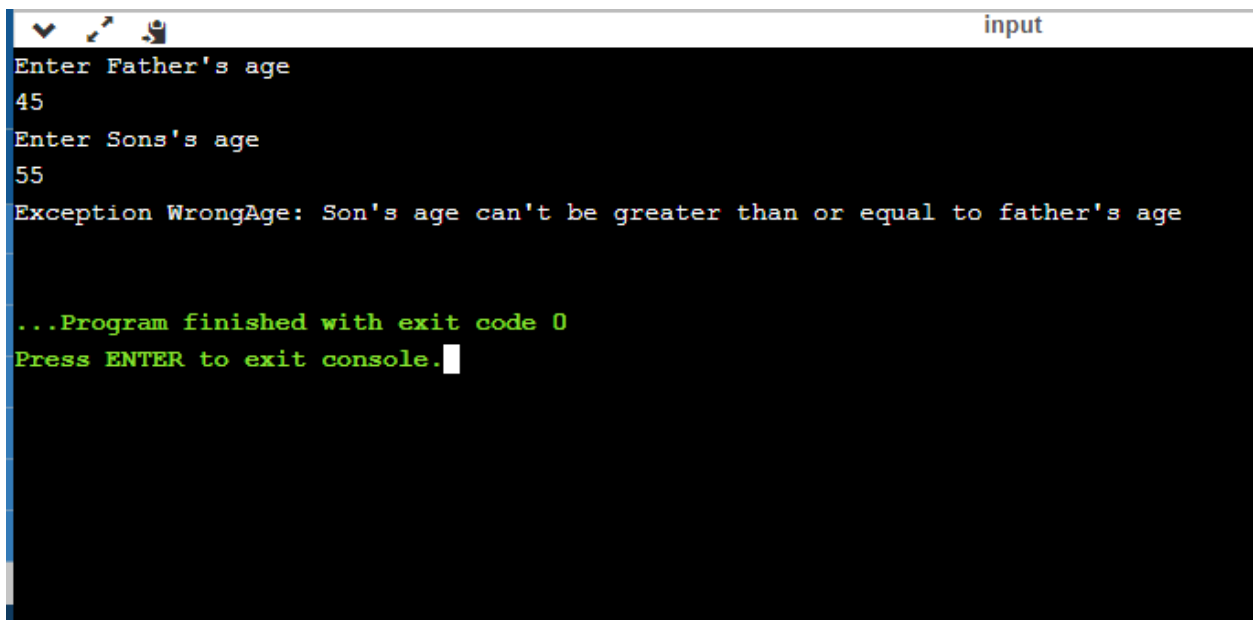
## Output-

```
input
Enter Father's age
55
Enter Sons's age
55
Exception WrongAge: Son's age can't be greater than or equal to father's age

...Program finished with exit code 0
Press ENTER to exit console.
```

```
Enter Father's age
56
Enter Sons's age
12
Father's Age: 56
Son's Age: 12

...Program finished with exit code 0
Press ENTER to exit console.
```



```
input
Enter Father's age
45
Enter Sons's age
55
Exception WrongAge: Son's age can't be greater than or equal to father's age

...Program finished with exit code 0
Press ENTER to exit console.
```

## 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.

*classmate*  
Date \_\_\_\_\_  
Page \_\_\_\_\_

WEEK 11: 8-12-2020 Mallika Prasad IBM19CS081

LP #9 WAP which creates 2 threads, one thread displaying 'BMS college of engineering' every 10 seconds and another displaying 'CSE' every 2 seconds

```
class NewThread implements Runnable {  
    String name;  
    int time;  
    Thread t;  
  
    NewThread (String threadname, int threadtime) {  
        name = threadname;  
        time = threadtime;  
        t = new Thread (this, name);  
        System.out.println ("New thread: " + t);  
        t.start();  
    }  
  
    public void run () {  
        try {  
            for (int i=1; i<=7; i++) {  
                System.out.println (name);  
                Thread.sleep (time);  
            }  
        } catch (InterruptedException e) {  
            System.out.println (name + "interrupted");  
        }  
    }  
}
```

```
System.out.println (name + "quitting");  
}  
}
```

```
class Main {  
    public static void main (String args[]) {  
        new NewThread ("CSE", 2000);  
        new NewThread ("BMS College of Engineering", 10000);  
    }  
}
```



## Output-

```
CT:Thread[NThread,5,main]
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE Thread quitting
BMS College of Engineering
BMS College of Engineering
BMS Thread quitting

...Program finished with exit code 0
Press ENTER to exit console.
```

## LAB 10-

*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.*

## Observation-

P#10: WAP that creates a user interface to perform integer divisions. The user enters 2 numbers in the textfields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 and Num2 were not an integer, the program would throw an Arithmetic Exception Display the exception in a message dialogue box

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
import javax.swing.*;
```

```
public class division extends Frame implements ActionListener {
```

```
    TextField n1, n2, res;
```

```
    Label l1, l2, lres;
```

```
    Button b;
```

```
    public division() {
```

```
        setLayout(new FlowLayout());
```

```
        label l1 = new Label("NUMBER1", Label.RIGHT);
```

```
        label l2 = new Label("NUMBER2", Label.RIGHT);
```

```
        label lres = new Label("RESULT", Label.RIGHT);
```

```
        n1 = new TextField(12);
```

```
        n2 = new TextField(8);
```

```
        res = new TextField(10);
```

```
b = new Button("DIVIDE");  
add(n1);  
add(n2);  
add(n1);  
add(n2);  
add(b);  
add(lus);  
add(lus);  
b.addActionListener(this);  
addWindowListener(new WindowAdapter());  
}
```

```
public void actionPerformed(ActionEvent ae) {  
    if (ae.getSource() == b) {  
        try {  
            int num1 = Integer.parseInt(n1.getText());  
            int num2 = Integer.parseInt(n2.getText());  
            int num3 = num1 / num2;  
            lus.setText(String.valueOf(num3));  
        } catch (NumberFormatException ne) {  
            JOptionPane.showMessageDialog(this, ne, "ERROR", JOptionPane.ERROR_MESSAGE);  
        } catch (ArithmeticException a) {  
            JOptionPane.showMessageDialog(this, a, "ERROR", JOptionPane.ERROR_MESSAGE);  
        }  
    }  
}
```

}

}

}

```
public static void main (String args[]) {
```

```
    division i = new division();
```

```
    i.setSize(new Dimension (700, 300));
```

```
    i.setTitle ("DIVISION OF TWO INTEGERS");
```

```
    i.setVisible (true);
```

```
}
```

```
class WindowAdapter1 extends WindowAdapter {
```

```
    public void windowClosing (WindowEvent we) {
```

```
        System.exit(0);
```

```
    }
```

```
}
```

```
}
```



## Output-

