

# **WEEK1**

## Program 7

DATE: \_\_\_\_\_  
PAGE: \_\_\_\_\_

```
import java.util.Scanner;
class quad
{
    public static void main (String args[])
    {
        int a; int b; int c;
        double d;
        double r1, r2;
        Scanner s1 = new Scanner (System.in);
        System.out.println ("Enter the values of");
        System.out.println ("a, b, c");
        a = s1.nextInt();
        b = s1.nextInt();
        c = s1.nextInt();
        d = b*b - 4*a*c;
        if (a == 0)
    }
```

System.out.println ("The equation is not quadratic");

}  
else if (d == 0)

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

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

}  
System.out.println (r1);

}  
else if (d > 0)

}  
System.out.println ("The roots are  
real and distinct");

$$n_1 = \frac{(-b + \text{Math.sqrt}(d))}{(2*a)}.$$

$$n_2 = \frac{(-b - \text{Math.sqrt}(d))}{(2*a)}.$$

System.out.println("n1=" + "n2");

else

{

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

$$n_1 = -b / (2*a);$$

$$n_2 = \text{Math.sqrt}(\text{Math.abs}(d));$$

System.out.println("The roots are " + "n1" + "n2");

$$" + n_1 + " + " + n_2 + " + " + \text{Math.sqrt}(d) +$$

$$" and " + n_1 + " + " + n_2 + " + " + \text{Math.sqrt}(d);$$

{

}

Q/P

Enter the coefficient a,b,c

0

1

2

Invalid input for d Enter the coefficients a,b,c

Enter the coefficients a,b,c

1

2

3

1

2

The roots are real & equal  
roots are  $r_1 = r_2 = -1.0$

The roots have no real  
solution & are imaginary  
 $-1.0 + i1.41421356$

$1.41421356 - i1.41421356$

Enter the coefficients a,b,c

1

4

The roots are real &  
the distinct roots are  $2679491914$

```
 Command Prompt
Imaginary roots.Roots are -0.5+i0.8660254037844386
C:\Users\bmscecse\Desktop\1BM21CS030>javac Quadratic.java
C:\Users\bmscecse\Desktop\1BM21CS030>java Quadratic.java
Enter coefficients
1 1 1
Imaginary roots.Roots are -0.5+i0.8660254037844386and -0.5-i0.8660254037844386
C:\Users\bmscecse\Desktop\1BM21CS030>java Quadratic.java
Enter coefficients
1 2 3
Imaginary roots.Roots are -1.0+ii.4142135623730951and -1.0-ii.4142135623730951
C:\Users\bmscecse\Desktop\1BM21CS030>java Quadratic.java
Enter coefficients
2 5 2
Roots are real and distinct. Roots are -0.5 and -2.0
C:\Users\bmscecse\Desktop\1BM21CS030>java Quadratic.java
Enter coefficients
0 1 2
Not a quadratic equation
C:\Users\bmscecse\Desktop\1BM21CS030>java Quadratic.java
Enter coefficients
1 2 1
Roots are equal and is equal to -1.0
C:\Users\bmscecse\Desktop\1BM21CS030>
```

## **WEEK 2**

```
import java.util.Scanner  
class Book
```

```
String name, usn;  
int marks[5] = new int[5];  
int credits[5] = new int[5];  
void input()  
{  
    Scanner s = new Scanner(System.in);  
    System.out.print("Enter your name:");  
    name = s.nextLine();  
    System.out.print("Enter your usn:");  
    usn = s.nextLine();  
    System.out.print("Enter the marks  
of each subject");  
    for (int i = 0; i < 5; i++)  
    {  
        marks[i] = s.nextInt();  
    }  
    System.out.print("Enter the credits  
for each subject");  
    for (int j = 0; j < 5; j++)  
    {  
        credits[j] = s.nextInt();  
    }  
}
```

void display()

{  
system.out.println("Name: " + name);  
system.out.println("USN: " + usn);  
for (int i = 0; i < 5; i++)

system.out.println("Marks of subject  
+ (i+1) " marks[i]);

System.out.println("No of subjects  
the subject above is "+  
marks[1]);

}

void calc()

{  
int gr.point[5] = new int[5];

gr.sgpa = 0;

gr.spm = 0;

float tot = 0;

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

{ (marks[i]) = 90

gr.point[i] = 10.

else if (marks[i] >= 80)

gr.point[i] = 8;

else if (marks[i] >= 70)

gr.point[i] = 7;

else if (marks[i] >= 60)

gr.point[i] = 6;

else if (marks[i] >= 50)

gr.point[i] = 5;

else if (marks[i] >= 40)

gr.point[i] = 4;

due of marks (if  $c[i] < 0$  marks > 0)  
 else  $g1.mark[i] = 0$

system.out.println ("Invalid input for  
 $g1.mark[i] = 0 + " subject$   
 $\{ \sum_{i=0}^{n-1} credits[i]$ )

$res = (float) sgpa / sum$   
 system.out.println ("SGPA = " + res)

class sgpa

public static void main (String args)

{  
 student s1 = new student();  
 s1.input();  
 s1.display();  
 s1.calc();  
 }

IP.

Enter your name:

~~Rajeshwar J~~

Enter your USN.

40

Enter the marks of each subject

66

77

85

56

87

~~import java.util.Scanner;~~  
~~class Book~~  
~~String name, author;~~

Enter the no. of credits for each subject

4

3

3

1  
3

Name: Bhardwaj

DN: 38

marks of subject 1 = 66

no of credits for subj above = 4

marks of subject 2 = 74

no of credits of subj above = 3

marks of subject 3 = 88

no of credits for subj above = 3

marks of subject 4 = 86

no of credits for subj above = 3

marks of subject 5 = 87

no of credits for subj above = 3

SGP = 8.0

```
C:\Users\BMSCECSEIL74\Desktop>javac LAB2.java
C:\Users\BMSCECSEIL74\Desktop>java lab_2
Enter the number of courses
9
Enter the credits of the courses:
4
1
1
1
1
3
3
3
3
Enter your usn,name
1BM21CS019
AMSHU G M
Enter your marks in each subject
enter the marks obtained in 0 course (credits= 4) :90
enter the marks obtained in 1 course (credits= 1) :98
enter the marks obtained in 2 course (credits= 1) :98
enter the marks obtained in 3 course (credits= 1) :84
enter the marks obtained in 4 course (credits= 1) :86
enter the marks obtained in 5 course (credits= 3) :56
enter the marks obtained in 6 course (credits= 3) :87
enter the marks obtained in 7 course (credits= 3) :87
enter the marks obtained in 8 course (credits= 3) :87

-----
usn:1BM21CS019
name:AMSHU G M
sgpa:8.85
```

## **WEEK 3**

~~WAP to create class~~  
Program-3

```
import java.util.Scanner  
class Book
```

```
{  
    String name, author;  
    double price;  
    int numPages;  
    Book();  
}
```

```
name = "", author = "";  
price = 0.0, numPages = 0;
```

```
void print()  
{
```

```
Scanner s = new Scanner(System.in);  
System.out.print("Enter the name of  
the book");
```

```
name = s.nextLine();  
System.out.print("Enter author of the  
book");
```

```
author = s.nextLine();  
System.out.print("Enter the price of the  
book");
```

```
price = s.nextDouble();  
System.out.print("Enter the no of  
pages");
```

```
numPages = s.nextInt();
```

public string to string()

return ("Name: " + name + " Author: " +  
author + "\n Price: " + price + "  
\n Number of pages: " +  
numPages + "\n");

}  
class Book main

{  
public static void main (String args)

Scanner s = new Scanner (System.in);  
System.out.print ("Enter the no of books")  
int n = s.nextInt();

Book books[] = new Book [n];  
for (int i=0; i<n; i+1){

books[i] = new book();

books[i].info();  
System.out.print ("Book details: ")  
System.out.print ("books[i].tostring()")

}  
}

O/P:

DATE

PAGE

Enter the number of books:  
1

Enter the name of the book:  
Power of thirty

Enter the author of the book  
James

Enter the price of the book  
450

Enter the number of pages:  
200

Book details

Name: Power of thirty

Author: James

Number of pages: 200

```
C:\Users\bmscse\Desktop\1BM21CS022>java Lab_3
Enter number of books to be created
2
Enter book name
java
Enter book author
person p
Enter number of pages and price of book
100 1000
Enter book name
ooj
Enter book author
person m
Enter number of pages and price of book
20 200
Book entry number 1
Title: java
Author: person p
Number of pages: 100
Price: 1000.0
Book entry number 2
Title: ooj
Author: person m
Number of pages: 20
Price: 200.0
```

# WEEK 4

### Program-4

WAP to create an abstract class called ship that contains two objects and an abstract method named printArea(). Provide three classes named rectangle, triangle and circle such that each one of the classes extends the class ship. Each one of the classes contains only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
import java.awt.*;
abstract class Shape {
    }
```

```
    draw() {}
```

```
    int h, b;
    abstract void printArea();
```

```
class Rectangle extends Shape
{
```

```
    Scanner s = new Scanner(System.in);
    void printArea()
```

```
System.out.println("Enter length and width of
rectangle");
    h = s.nextInt();
    b = s.nextInt();
```

```
    System.out.println("Area of Rectangle is
        + b * h);
```

```
    Rectangle() {}
```

class triangle extends shape

{  
Scanner s = new Scanner (System.in);  
void printArea ()

System.out.println ("Enter height and  
base of rectangle");  
h = s.nextInt();  
b = s.nextInt();

System.out.println ("Area of Triag. =  
 $\frac{1}{2} \times h \times b$ ");

{  
triangle ()  
}

class circle extends shape

{  
Scanner s = new Scanner (System.in);  
void printArea ()

System.out.println ("Enter the radius of  
circle");  
r = s.nextInt();

System.out.println ("Area of circle =  
 $\pi r^2$ ");

{  
circle ()  
}

class main

{  
public static void main (String args[])

rectangle r = new rectangle();

r.printArea();

triangle t = new triangle();

t.printArea();

circle c = new circle();

c.printArea();

}

Output

Enter length & breath of rectangle

3

Area of rectangle is 6

Enter height & base of triangle

3

3

Area of triangle is 7.5

Enter the radius of circle

4

Area of circle is 50.24.

```
Enter the shape
```

```
1.Rectangle
```

```
2.Triangle
```

```
3.Circle
```

```
4.Exit
```

```
1
```

```
Enter breadth and height
```

```
2 4
```

```
Area of the rectangle is 8
```

```
2
```

```
Enter side length and height
```

```
2 4
```

```
Area of the triangle is 4
```

```
3
```

```
Enter radius
```

```
1
```

```
Area of the circle is 3.141592653589793
```

```
4
```

# WEEK 5

## Lab program - 5

WAP to create a class Bank that maintains two kinds of account for its customer, one called savings account and the other current account. The savings account provides combined deposit and withdrawal facilities but no cheque book facility. The current account provides cheque book facilities but no deposit. Current account holder should also maintain a minimum balance and if the balance falls below this limit, a service charge is imposed. Create a class Account that stores customer name, age, no. and type of account. From this derive Savings account and current account. In the Savings account class, the withdraw function is specific to 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 balance. Deduct for minimum balance, impose penalty and update the balance.

```
import java.util.Scanner  
import java.lang.Math
```

class Account

```
{  
    String name = new String();
```

```
    int accno;
```

```
    double bal;
```

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

```
    void set()
```

```
{
```

```
    System.out.println("Enter customer name")  
    name = s.nextLine();
```

```
    System.out.println("Enter " + name + "'s  
    account number");
```

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

```
    System.out.println("Enter balance amount");
```

```
    bal = s.nextDouble();
```

```
    void display()
```

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

```
    System.out.println("Your Acc number: " + accno);
```

```
    System.out.println("Customer Account Balan:  
    " + bal);
```

```
}
```

```
Account() { }
```

```
{
```

class savacc extends account

{  
Scanner sc = new Scanner (System.in);  
savacc()

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

{  
void deposit ()

{  
not dij  
double amt;  
System.out.print ("Please enter amount to deposit");  
ch = s.nextInt();  
if (ch == 1)

System.out.print ("Enter amount to be deposited");  
amt = s.nextInt();  
bal = bal + amt;

{  
else  
System.out.print ("Invalid input")

{  
void m ()

System.out.print ("Enter rate of interest");  
double r = s.nextDouble();

System.out.print ("Enter number of times  
interest applied per the period");  
int n = s.nextInt();

System.out.print ("Enter size of the  
period");

Double  $x = (1 + (A / 100))$   
Double  $ci = \text{bal} * \text{math. how}(x, y);$   
System.out.println("Interest amount is " + ci);  
System.out.println("Balance amount without interest  
is " + bal);

System.out.println("Available balance after  
updtg s & ci");

void wd()

{

S. O. P ("Enter amount to be withdrawn");

w = S. nextDouble();

if (balance > w) {

balance = balance - w;

S. O. P ("New Balance is " + balance);

}

else {

if (choice == 1) {

sav\_acc.sav = new SavAcc();

sav.set();

sav.deposit();

sav.withdraw();

sav.Interest();

}

else if (choice == 2)

curr\_acc.curr = new CurrAcc();

curr.set();

curr.deposit();

curr.withdraw();

curr.interest();

```
Press
1. for Savings account
2. for Current account
1
Facilities available are :
1. Withdrawal
2. Compound Interest
3. No Cheque
Enter your name :
qwe
Enter Account number :
1
Enter balance:
1000
Name : qwe
Account number : 1
Balance : 1000.0
Enter 1 to deposit :
1
Enter the amount to deposit :
1000
Enter rate of interest :
5
Enter frequency of interest applied per time period :

1
Enter time periods :
1
Interest amount=100.0
Balance amount without interest is 2000.0
Available balance after updating is : 2100.0
Enter 1 to withdraw :
1
Enter the amount you want :
100
Available balance is : 2000.0
```

# WEEK 6

Week-6

1. Create a class Customer with the following specifications.

Private members

Customer no, Customer name, City, Price, Total Price,  
Discount, Net price.

Method: Public member:

1. A parameterized constructor to assign  
initial

2. Input() - to read data members  
call calculate()

3. calculate() - to calculate discount  
according to Total Price  
and Net Price.

$$\text{Total Price} = \text{Price} * \text{Qty}$$

$$\text{Total Price} >= 50000 - \text{Discount 25\% of 7K}$$

$$\text{Total Price} >= 25000 - \text{Discount 10\% of 7K}$$

$$\text{Net Price} = \text{Total Price} - \text{Discount}$$

n. show() - to display customer details.

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

```
class Customer {
```

```
    private int cust_no;
```

```
    private String cust_name;
```

```
    private int quantity;
```

```
    private float price;
```

```
    private float total_price, discount, net_price;
```

```
    customer() {}
```

```
customer (int no, String name, int qua, float pri) {
```

```
    cust_no = no;
```

```
    cust_name = name;
```

```
    quantity = qua;
```

```
    price = pri;
```

```
    total_price = quantity * price;
```

```
}
```

```
void input ()
```

```
{
```

```
    Scanner s = new Scanner (System.in)
```

```
    System.out.print ("Enter customer number, customer name,
```

```
    item quantity, item price")
```

```
    cust_no = s.nextInt();
```

```
    cust_name = s.next();
```

```
    quantity = s.nextInt();
```

```
    price = s.nextFloat();
```

```
    total_price = quantity * price;
```

```
.
```

void car\_discount() {
 if (total\_price >= 50000) {
 discount = (float) (total\_price \* 0.25);
 } else if (total\_price >= 25000) {
 discount = (float) (total\_price \* 0.1);
 } else {
 discount = 0;
 }
 }

net price = total\_price - discount;
 }

void show() {
 System.out.println ("Customer details");
 System.out.println ("Name " + customer\_name + " without " +
 "or " + customer\_name + " with name " + customer\_name +
 " grant");
 System.out.println ("Total price : " + float +
 " in Total price " + total\_price +
 " in Net price " + net\_price + " in Net price");
 }
}

class Test {
 public static void main (String args[]) {
 Scanner sc = new Scanner (System.in);
 System.out.print ("Enter the no of objects to be entered:");
 int n = sc.nextInt();

for (int i = 1; i <= n; i++) {
 Car car = new Car();
 car.getInputs();
 car.show();
 }
 }
 }
}

Customer CTC -> New customer [CJ]  
for  $i \in 0 \text{ to } n; i+1$   
 $\quad$   
System adds node ("False customer number exists  
false, stem quantity, from price")  
 $\quad$   
 $\quad$  if no = 1.insert(i);  
 $\quad$  else name = S.name();  
 $\quad$  ~~not allow = 2.insert()~~  
 $\quad$  first ID = S.name() + i  
 $\quad$   $C[i] = \text{new customer } (n, \text{name}, \text{true}, \text{false})$   
 $\quad$   $C[i].oldDiscard(1)$   
 $\quad$   $\quad$   
 $\quad$  for  $i \in 0; i < i-1; i+1$   
 $\quad \quad$  if  $C[i].name == C[i+1].name$   
 $\quad \quad \quad$   $C[i].oldDiscard(1)$   
 $\quad \quad \quad$   $S$   
 $\quad \quad \quad$   $\quad$   
 $\quad \quad \quad$   $\quad$

```
3
Enter customer number,customer name,item quantity,item price
1 qwe 5 5000
Enter customer number,customer name,item quantity,item price
2 rty 3 10000
Enter customer number,customer name,item quantity,item price
3 yui 6 20000
-----Customer Details-----
```

```
Customer number: 1
Customer name: qwe
Quantity: 5
Item price: 5000.0
Total price: 25000.0
Discount: 2500.0
Net price: 22500.0
```

```
-----Customer Details-----
```

```
Customer number: 2
Customer name: rty
Quantity: 3
Item price: 10000.0
Total price: 30000.0
Discount: 3000.0
Net price: 27000.0
```

```
-----Customer Details-----
```

```
Customer number: 3
Customer name: yui
Quantity: 6
Item price: 20000.0
Total price: 120000.0
Discount: 30000.0
Net price: 90000.0
```

WEEK 6(2)

Q8:

8  
Week 6 (2)

Develop a Java program to create a class Patient with data members pt-id, pt-name, pt-age, dob.

The program should include the following functionalities

- Accept patient details
- Accept a patient id and display his/her details.
- Accept the name of the doctor and display the name of all patients treated by him/her.

import java.util.Scanner;

```
class Patient {  
    int p-id;  
    String p-name;  
    int p-age;  
    String dob;
```

void set() {

System.out.print ("Enter patient ID, name,  
age and attorney details: ");

Scanner s = new Scanner (System.in);  
p-id = s.nextInt();

p-name = s.next();

p-age = s.nextInt();

dob = s.next();  
}

void display() {

System.out.println("Enter patient details - ");  
System.out.println("Patient ID: " + patientID);  
PatientName = scanner.nextLine();  
"Patient name: " + PatientName + " in key: " + age;  
"Age: " + age;

{

Class Lab - 62 {

public static void main (String args) {

Scanner s = new Scanner (System.in);  
int n = s.nextInt();

patientID = new patient();  
patientID = new patient();

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

patientID = new patient();  
patientID.set();

{

int choice;

do {

System.out.println("Enter choice\n1. Patient  
details\n2. Patients grouped by date  
3. Exit");

choice = s.nextInt();

if (choice == 1) {

System.out.println("Enter patient ID: ");

int id = s.nextInt();

if (id >

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

{  
if ( $p[i].p\_id = id$ )

{  
 $p[i].oddby(i)$   
break;

{  
if ( $i == n$ ) {

System.out.println("Kontrol done");

{  
else if ( $choice == 2$ ) {

System.out.println("Enter domain");

String d = s.nextLine();

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

if ( $p[i].d == d$ ) {

System.out.println( $p[i].p = \text{new}$ );

break;

{  
if ( $i == n$ ) {

System.out.println("Porta done");

choice = 3;

```
Enter the number of patients:  
2  
Enter patient id, name, age and doctor name:  
1 qw 23 wqe  
Enter patient id, name, age and doctor name:  
2 er 34 qwr  
1:Enter patient ID  
2:Enter Doctor name  
3:Exit  
1  
Enter patient ID:  
1  
1  qw  23  wqe  
1:Enter patient ID  
2:Enter Doctor name  
3:Exit  
2  
Enter dooctor name:  
wqe  
1  qw  23  wqe  
1:Enter patient ID  
2:Enter Doctor name  
3:Exit  
3
```

# WEEK 8

## lab-8

WAP creating two thread one thread  
displaying "BMS college of Engineering  
once every 1 sec and another displaying  
"CSE" one every two seconds.

class Thread implements Runnable  
{  
 String n;  
 int interval, times;  
 Thread thread;

Thread (String name, int intervals, int)  
{

n=name;  
interval=intervals;

times=1;

thread=new Thread (this, name)  
thread.start();

}

public void run(){}

try {

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

{

System.out.println (n);

Thread.sleep (interval);

}

catch (extraction etc)

system and partly (etc etc)

DATE:

PAGE:

77

class lab 83

bubble state word man (theory app 223)

new Bhorad ( "BMS college of engg")

1000 ft

new Bhorad (4 (SE, 2000, 15))

{  
}{  
}{  
}

O/P

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CE

CE

CE

CE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

WEEK 6(3)

### Week 6 (3)

Create an Abstract class calculate  
which has three double members  
- say x, y and result include  
a method calc. derive three classes  
from calculate which performs any  
three arithmetic operations on the  
two variables x & y and assign  
the result to result  
make appropriate declarations and  
definitions

import java.util.Scanner;

abstract class calculate  
double x, y, result;  
abstract void calc();  
{ }

class Addtors extends calculate  
void calc () {  
System.out.print ("Enter two  
numbers x and y for addition: ");  
Scanner s = new Scanner (System.in);  
x = s.nextDouble();  
y = s.nextDouble();  
result = x + y;  
System.out.println ("Addition of " + x + "  
" + y + " is " + result);  
}

5  
addition (7/33)

DATE:

PAGE:

class subtraction extends calculate {

void calc () {

System.out.print("Enter two no. n  
and y for sub: ")

Scanner s = new Scanner (System.in);

n = s.nextDouble();

y = s.nextDouble();

System.out.println("result = " +

System.out.print(" (" + n - y + ") is : " + result));

5  
subtraction (7/33)

class Multiplication extends calculate {

void calc () {

System.out.print(" Enter two  
numbers x and y for multiplication: ")

Scanner s = new Scanner (System.in);

x = s.nextDouble();

y = s.nextDouble();

result = x \* y;

S.O. P ("Multiplication of " + x + "  
" + y + " is : " + result));

5  
multiplication (7/33)

class Division extends calculate

{  
void calc()  
{

System.out.println ("Enter two reals numbers");  
Scanner ss = new Scanner (System.in);

x = ss.nextDouble();

y = ss.nextDouble();

result = x/y;

System.out.println ("Division of " +  
"and " + y + " is: " + result);

}

division.cs

class True {

public static void main (String args[]){

Addition A = new Addition();

A.calc();

Subtraction S = new Subtraction();  
S.calc();

Multiplication M = new Multiplication();

M.calc();

Division D = new Division();

D.calc();

8/15

DATE:

PAGE:

Enter two numbers  $x$  and  $y$  for addition:

Addition of  $1.0$  and  $2.0 = 3.0$

Enter two numbers  $x$  and  $y$  for subtraction:

Subtraction of  $1.0$  and  $2.0 = -1.0$

Enter two numbers  $x$  and  $y$  for multiplication:

Multiplication of  $1.0$  and  $2.0 = 2.0$

Enter two numbers  $x$  and  $y$  for division:

Division of  $1.0$  and  $2.0 = 0.5$

```
Enter two numbers x and y for addition :  
1 2  
Addition of 1.0 and 2.0 is : 3.0  
Enter two numbers x and y for subtraction :  
1 2  
Subtraction of 1.0 and 2.0 is : -1.0  
Enter two numbers x and y for multiplication :  
1 2  
Multiplication of 1.0 and 2.0 is : 2.0  
Enter two numbers x and y for division :  
1 2  
Division of 1.0 and 2.0 is : 0.5
```

# WEEK 7

Week 7

Write a program that demonstrates  
handling of exceptions in inheritance tree  
using a base class called 'father'  
and derived class called 'son'  
which extends the base class.  
In father class, implement a constructor  
which takes age that covers both  
father and sons age and throws  
an exception if son's age is greater  
than father's age.

import java.util.Scanner;  
class AgeException extends Exception  
{  
 public String toString()  
 {  
 return "Negative age can't be  
 accepted";  
 }  
}

class AgeException extends Exception  
{  
 public String toString()  
 {  
 return "Son can't be older than  
 father";  
 }  
}

class Father {

int father\_age;  
Father (int a) throws invalid Father  
{  
 father\_age = a;  
 if (father\_age < 0)  
 throw new WrongAgeException();  
}

class Son extends Father {

int son\_age;  
Son (int a, int b)  
{

super(a);

son\_age = b;

if (son\_age < 0)

throw new WrongAgeException();

if (son\_age >= father\_age)

throw new AgeException();

{ }  
{ }

class lab 7

{  
Public static void main (String args)

}  
by {

scanners = new Scanner (System.in);  
System.out.println ("Enter father's age")  
sons age})

int x = s.nextInt();

int y = s.nextInt();

int z = new Scanner (System.out);  
System.out.println ("Mother's age")  
old and son > 18 years old,  
so father age, so son age

}  
catch (AgeException e) {

System.out.println (e);

}  
catch (AgeException e) {

System.out.println (e);

}  
catch (Exception e) {

System.out.println ("Enter valid input")

}  
}

97

Enter father and son's ages

35

15

Father is 35 years old and son is 15 years old

Enter father and son's ages

-98

70

negative ages not accepted

Enter father's and son's ages

37

- 9

negative ages ~~can't~~ be accepted

Enter father and son's ages

23

52

son must be older than father

```
Enter father's and son's ages
```

```
35 15
```

```
Father is 35 years old and son is 15 years old
```

```
Enter father's and son's ages
```

```
-98 90
```

```
Negative age can't be accepted
```

```
Enter father's and son's ages
```

```
34 -9
```

```
Negative age can't be accepted
```

```
Enter father's and son's ages
```

```
34 -9
```

```
Negative age can't be accepted
```

# WEEK 8

## lab-8

WAP creating two thread one thread  
displaying "BMS college of Engineering  
once every 1 sec and another displaying  
"CSE" one every two seconds.

class Thread implements Runnable  
{  
 String n;  
 int interval, times;  
 Thread thread;

Thread (String name, int intervals, int)  
{

n=name;  
interval=intervals;

times=1;

thread=new Thread (this, name)  
thread.start();

}

public void run(){}

try {

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

{

System.out.println (n);

Thread.sleep (interval);

}

catch (extraction etc)

system and partly (etc etc)

DATE:

PAGE:

77

class lab 83

bubble state word man (theory app 223)

new Bhorad ("BMS college of engg")

1000 ft

new Bhorad (4 (SE, 2000, 15))

{  
}{  
}{  
}

O/P

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

BMS college of Engg

CSE

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

CSE

CE

CE

CE

CE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE

BMS College of Engineering

CSE

CSE

CSE

CSE

CSE