

## Lab Program 1:

Develop a Java program that prints all real solutions to the quadratic equation  $ax^2+bx+c = 0$ . 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.

## SOURCE CODE:

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

public class QuadEq {
    public static void main (String ss[]) {
        Scanner x = new Scanner(System.in);
        System.out.print("Enter the value of a: ");
        double a;
        do {
            System.out.println("make sure it is not 0");
            a=x.nextDouble();
        }while(a==0);

        System.out.print("Enter the value of b: ");
        double b=x.nextDouble();
        System.out.print("Enter the value of c: ");
        double c=x.nextDouble();

        double d=b*b-(4*a*c);
        double r1,r2;

        if(d>0) {
            r1=(-b + Math.sqrt(d))/(2*a);
            r2=(-b - Math.sqrt(d))/(2*a);
            System.out.println("Roots are real and distinct");
            System.out.println("Roots are: "+r1+" and "+r2);
        }

        else if(d==0) {
            r1=r2=-b/(2*a);
            System.out.println("Root is real and unique");
            System.out.println("Root is: "+r1);
        }
        else {
            r1=-b/(2*a);
            r2=Math.sqrt(Math.abs(d))/(2*a);
            System.out.println("There are no real solutions");
            System.out.println("Roots are imaginary and distinct");
            System.out.println("Root 1 is: "+r1+" + i"+r2);
            System.out.println("Root 2 is: "+r1+" - i"+r2);
        }
    }
}
```

## WRITTEN CODE:

### LAB-program 1

### Quadratic Equation

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

public class QuadEq {
    public static void main (String ss[]) {
        Scanner x = new Scanner (System.in)
        System.out.println ("Enter value of a: ");
        double a;
        do {
            System.out.println ("make sure it is not 0");
            a = x.nextnextDouble ();
        } while (a == 0);
        System.out.println ("Enter the value of b: ");
        double b = x.nextDouble ();
        System.out.println ("Enter the value of c: ");
        double c = x.nextDouble ();

        double d = b*b - (4*a*c);
        double r1, r2;

        if (d > 0) {
            r1 = (-b + Math.sqrt(d))/(2*a);
            r2 = (-b - Math.sqrt(d))/(2*a);
            System.out.println ("Roots are Real and Distinct");
            System.out.println ("Roots are: "+r1+" and "+r2);
        }
```

~~correct~~

else if (d == 0) {

$$r_1 = r_2 = -b / (2 * a);$$

System.out.println("Roots are real & distinct");

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

}

else {

$$r_1 = -b / (2 * a);$$

$$r_2 = \text{Math.sqrt}(\text{Math.abs}(d) / (2 * a));$$

System.out.println("There are no real solutions");

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

System.out.println("Root 1 is: " + r1 + " + i" + r2);

System.out.println("Root 2 is: " + r1 - "i" + r2);

}

}

}

Test cases:

① 1, -2, 1

② 1, 5, 3

③ 3, 1, 4

## OUTPUT (including test cases):

```
PS C:\Users\anosh\OneDrive\Desktop\java practice> javac QuadEq.java
PS C:\Users\anosh\OneDrive\Desktop\java practice> java QuadEq
Enter the value of a: make sure it is not 0
0
make sure it is not 0
1
Enter the value of b: -2
Enter the value of c: 1
Root is real and unique
Root is: 1.0
PS C:\Users\anosh\OneDrive\Desktop\java practice> java QuadEq
Enter the value of a: make sure it is not 0
1
Enter the value of b: 5
Enter the value of c: 3
Roots are real and distinct
Roots are: -0.6972243622680054 and -4.302775637731995
PS C:\Users\anosh\OneDrive\Desktop\java practice> java QuadEq
Enter the value of a: make sure it is not 0
3
Enter the value of b: 1
Enter the value of c: 4
There are no real solutions
Roots are imaginary and distinct
Root 1 is: -0.16666666666666666 + i2.798809270624444
Root 2 is: -0.16666666666666666 - i2.798809270624444
PS C:\Users\anosh\OneDrive\Desktop\java practice>
```



## Lab Program 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.

## OUTPUT:

```
C:\Users\anosh\OneDrive\Desktop\java practice>javac smain.java

C:\Users\anosh\OneDrive\Desktop\java practice>java smain
Enter no. of subjects:
6
enter usn, name, credits, marks for subjects
1BM21CS024
Anoshor B Paul
1 2 2 3 3 4
85
92
75
87
90
93
USN: 1BM21CS024
Name: Anoshor B Paul
Marks:
85.0 92.0 75.0 87.0 90.0 93.0
Credits:
1 2 2 3 3 4
Your SGPA is: 9.466666666666667
```

## WRITTEN CODE:

### LAB Program 2

2/12/22

~~import~~

import java.util.Scanner;

class student {

String usn;

String name;

~~double~~ <sup>int</sup> credits[] = ~~new double[6]~~ <sup>new int[6]</sup>;

double marks[] = ~~new double[6]~~ <sup>new double[6]</sup>;

void getdata() {

Scanner x = new Scanner(System.in);

System.out.println("enter usn, name, credits, marks");

usn = x.~~next~~ nextLine();

name = x.~~next~~ nextLine();

for (int i=0; i<6; i++)  
credits[i] = x.next <sup>int</sup> ~~double~~ ();

~~marks~~  
for (int i=0; i<6; i++)  
marks[i] = x.next Double ();

}

void putdata() {

System.out.println("USN: " + this.usn);

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

for (int i=0; i<6; i++) {

System.out.print("Credit, Marks ");

System.out.println(" \* this.credits[i] + " " + this.marks[i], i);

}

y

// Default constructor

Student() {

System.out.println("no. of subjects: ");

Scanner s = new Scanner(System.in);

int n;

n = s.nextInt();

this.credits = new int[n];

this.marks = new double[n];

}

class smain {

public static void main (String ss[])

student s1 = new student();

s1. getsd();

s1. putsd();

s1. sgpa();

}

Test case: Enter no. of subjects  
> 6

IBM21CS024

Anashor B Paul

1 2 2 3 3 4

88 88 90 76 70 99

USN: IBM21CS024

Name: Anashor B. Paul

Marks: 88.0 88.0 90.0 76.0 70.0 99.0

Credits: 1 2 2 3 3 4

Your SGPA is : 9.0

```
void sqa() {
```

```
    double marks = 0
```

```
    double t-credits = 0;
```

```
    for (int i = 0; i < 6; i++) {
```

```
        if (this.marks[i] >= 90) {
```

```
            marks = marks + (10 * (this.credits[i]));
```

```
        } else if (this.marks[i] >= 80) {
```

```
            marks = marks + (9 * (this.credits[i]));
```

```
        } else if (this.marks[i] >= 70) {
```

```
            marks = marks + (8 * (this.credits[i]));
```

```
        } else if (this.marks[i] >= 60) {
```

```
            marks = marks + (8 * (this.credits[i]));
```

```
        } else if (this.marks[i] >= 50) {
```

```
            marks = marks + (7 * (this.credits[i]));
```

```
        } else if (this.marks[i] >= 40) {
```

```
            marks = marks + (6 * (this.credits[i]));
```

```
        } else {
```

```
            marks += 0;
```

```
        }  
        t-credits += this.credits[i];
```

```
    }  
    double sqa = (marks / t-credits);
```

```
    System.out.println("Your Sqa is " + sqa);
```



### Lab Program 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.

### OUTPUT:

```
C:\Users\anosh\OneDrive\Desktop\java practice>java bmain
Enter the number of books:
2
Type information of book 1
Enter the book name, author, price and number of pages:
Harry Potter
JK Rowling
1000
2300
Type information of book 2
Enter the book name, author, price and number of pages:
Tale of Two Cities
Charles Dickens
750
1500

Book 1Details: Name    Author    Price    no_of_pages
Harry Potter JK Rowling 1000.0 2300

Book 2Details: Name    Author    Price    no_of_pages
Tale of Two Cities Charles Dickens 750.0 1500
```

## WRITTEN CODE:

### LAB Program 3

2/12/22

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

```
class Book book {
```

```
    String name, author;
```

```
    double price;
```

```
    int num-pages;
```

```
    void book() {
```

```
        this.name = "This is a Book";
```

```
        this.author = null;
```

```
        this.price = 0.0;
```

```
        this.numPages = 0;
```

```
    }
```

```
    void setd() {
```

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

```
        System.out.println("Enter the book name,  
author, price, number of pages: ");
```

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

```
        this.author = ss.next();
```

```
        this.price = ss.nextDouble();
```

```
        this.numPages = ss.nextInt();
```

```
    }
```

```
    public String toString() {
```

```
        return (name + " " + author + " " + price + " " + numPages);
```

```
    }
```

```
void display() {
```

```
    System.out.println("Book name is : " + this.name);  
    System.out.println("Author's name is : " + this.author);  
    System.out.println("Price of book : " + this.price);  
    System.out.println("Book has " + num-pages + " pages");
```

```
}
```

```
}
```

```
class main {
```

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

```
        int n;
```

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

```
        System.out.println("Enter the number of books: ");
```

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

```
        book B books[] = new book [n];
```

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

```
            books[i] = new book();
```

```
        }
```

```
        int m = n;
```

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

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

```
    System.out.println("Type inform of book" + (i+1));  
    books[i].setd;
```

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

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

```
    System.out.println("Name Author Price Num-of-pages");
```

```
    System.out.println(books[i]); // toString y y y
```





# **LAB PROGRAM 4**

## **QUESTION:**

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.

## WRITTEN CODE

### Lab Program 4

④ Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide 3 classes named: Rectangle, Circle, Triangle, such that each of them extends the 'Shape' class. Each of them contains the printArea() method to print the area of the given shape.

```
import java.util.Scanner;

abstract class Shape Figure {
    int no-pama;
    double area;
    abstract void printArea();
    // abstract void calcPerimeter();
}
```

```
class Rectangle Shape extends Figure {
    int sides[] = new int[4];

    Rectangle() {
        System.out.println("Figure is a Rectangle");
        no-pama = 4;
        Scanner ss = new Scanner(System.in);
        System.out.println("Enter 2 adjacent sides:");
        for (int i = 0; i < 2; i++) {
            sides[i] = ss.nextInt();
            sides[i+2] = sides[i];
        }
    }

    void printArea() {
        area = sides[0] * sides[1];
        System.out.println("Area of Rectangle: " + area);
    }
}
```

```
class Triangle extends Figure {
```

```
    int sides = new int [3];
```

```
    Triangle() {
```

```
        System.out.println("Figure is a Triangle");
```

```
        no-parea = 3
```

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

```
        System.out.println("Enter the 3 sides: ");
```

```
        for (int i = 0; i < no-parea; i++) {
```

```
            sides[i] = ss.nextInt();
```

```
        }
```

```
        void printArea() {
```

```
            if (sides[0] + sides[1] <= sides[2] || sides[1] + sides[2] <= sides[0] || sides[0] + sides[2] <= sides[1])
```

```
                System.out.println("Invalid Input");
```

```
            else {
```

```
                area = (double) Math.sqrt(s * (s - sides[0]) * (s - sides[1]) * (s - sides[2]));
```

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

```
            }
```

```
    }
```

class Circle extends Figure {

int ~~size~~ radius;

Circle() {

System.out.println("Figure is a circle");

no-param = 1

Scanner ss = new Scanner(System.in);

System.out.println("Enter the radius:");

radius = ss.nextInt();

}

void printArea() {

area = 3.14 \* radius \* radius;

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

}

}

class main {

public static void main (String ss[]) {

Rectangle r1 = new Rectangle();

Triangle t1 = new Triangle();

Circle c1 = new Circle();

OUTPUT:

Figure is a Rectangle  
Enter the 2 adjacent sides:

2 3

Figure is a Triangle.  
Enter 3 sides

Figure is a circle  
Enter the radius  
5

Area of Rectangle is 6.0

Area of Triangle is 6.0

Area of Circle is 78.5



# OUTPUT:

```
C:\Users\anosh\OneDrive\Desktop\java practice>java fmain
Figure is a Rectangle
Enter the 2 adjacent sides:
2 3
Figure is a Triangle
Enter the 3 sides:
3 4 5
Figure is a circle
Enter the circle radius:
5
Area of Rectangle is 6.0
Perimeter of Rectangle is: 10.0
Area of Triangle is 6.0
Perimeter of Triangle is: 12.0
Area of circle is 78.5
Circumference is: 31.400000000000002
```

**QUESTION:**

Lab Program 5

- a. develop a Java program to create a class Bank that maintains 2 kinds of account for its customers, one called savings and the other current account. The savings account provides compound interest & withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account 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 & type of account. From this derive the classes cur-acct & 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 customers & update the balance.
  - (b) Display the balance
  - (c) Compute & deposit interest
  - (d) Permit withdrawal & update the balance
- check for min. balance, impose penalty if necessary & update the balance.

## WRITTEN CODE

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

```
class Account
```

```
class Account {
```

```
    String name;
```

```
    int type;
```

```
    long accno;
```

```
    double balance;
```

```
    void setA() {
```

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

```
        System.out.println("Enter customer name: ");
```

```
        name = x.nextLine();
```

```
        System.out.print("Enter account number: ");
```

```
        accno = x.nextLong();
```

```
        System.out.print("Enter bank balance: ");
```

```
        balance = x.nextDouble();
```

```
    }
```

```
    void display() {
```

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

```
        if (type == 1) {
```

```
            System.out.println("Acc type: Savings");
```

```
        } else {
```

```
            System.out.println("Acc type: Current");
```

```
        }
```

```
        System.out.println("Acc number no: " + accno);
```

```
        System.out.println("Current Balance: " + balance);
```

```
    }
```

```
void deposit() {
```

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

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

```
    double amt = x.nextDouble();
```

```
    balance = balance + amt;
```

```
}
```

```
}
```

```
class Sav-act extends Account {
```

```
    double interest;
```

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

```
    Sav-act() {
```

```
        type = 1;
```

```
}
```

```
    void interest() {
```

```
        int Hmny;
```

```
        float Hrate;
```

```
        System.out.println("Compound Interest details:");
```

```
        System.out.println("Enter time in years:");
```

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

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

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

```
        System.out.println("Interest will be compounded 5  
times a year");
```

```
        interest = balance * (Math.pow((1 + Hrate/5), (5 * Hmny)))
```

```
        balance += interest;
```

```
}
```



```
void withdraw() {
```

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

```
    double amt = s.nextDouble();
```

```
    if (balance > amt)
```

```
        balance -= amt;
```

```
    else
```

```
        System.out.println("Amt to be withdrawn greater  
        than balance");
```

```
}
```

```
class Curr_acct extends Account {
```

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

```
    double cheque-amount;
```

```
        cheque-amount;
```

```
    Curr_acct() {
```

```
        type = 2
```

```
    }
```

```
    void cheque() {
```

```
        System.out.println("Enter the cheque amt: ");
```

```
        cheque-amount = s.nextDouble();
```

```
        if (cheque-amount > balance - 5000) {
```

```
            System.out.println("Rs. 500 penalty imposition?   
            y or no ");
```

```
            String option = s.next();
```

```
            if (option.equals("y")) {
```

```
                balance = balance - cheque-amount - 500;
```

```
            else  
                System.out.println("Rs. " + cheque-amount + " is   
                out  
                balance -= cheque-amount;
```

```
void withdraw () {
```

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

```
    double amt = s.nextDouble();
```

```
    if (balance > amt)
```

```
        balance -= amt;
```

```
    else
```

```
        System.out.println("Amount greater than balance");
```

```
}  
}  
  
class Bank {
```

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

```
        int int op1, op2;
```

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

```
        System.out.println("1. Savings 2. Current ??");
```

```
        int q;
```

```
        q = next s.nextInt();
```

```
        if (q == 1) {
```

```
            SavAcct s1 = new SavAcct();
```

```
            while (true) {
```

```
                System.out.println("1. Set Details, 2. Display Details,  
                3. Withdraw, 4. Deposit, 5. Compound Interest 6. Exit");
```

```
                op1 = next s.nextInt();
```

```
                switch (op1)
```

```
                case (1): s1.setA(); break;
```

```
                case (2): s1.set display(); break;
```

```
                case (3): s1.deposit(); break;
```

```
                case (4): s1.compinterest(); break;
```

```
                case (5): s1.withdraw(); break;
```

```
                case (6): System System.exit(0);
```

else if (q == 2) {

    currAcct c1 = new currAcct();

    while (true) {

        System.out.println (1. Auth details & display details  
                                  3. deposit amt 4. cheque 5. withdraw 6. exit);

        op2 = s.nextInt();

        switch (op2) {

            case (1): c1.setA(); break;

            case (2): c1.display(); break;

            case (3): c1.deposit(); break;

            case (4): c1.cheque(); break;

            case (5): c1.withdraw(); break;

            case (6): System.exit(0);

        }

    }

}

# OUTPUT

```
C:\Users\anosh\OneDrive\Desktop\java practice>java Bank
1. Savings or    2. Current?
1
Enter the choice:
1 .Set the values for savings acc
2. display
3. deposit
4. Interest
5. Withdraw
6. exit
1
Enter customer name: Anoshor
Enter account number: 123024
Enter bank balance: 10000
Enter the choice:
1 .Set the values for savings acc
2. display
3. deposit
4. Interest
5. Withdraw
6. exit
4
Compound Interest details:
Enter time in years:
2
Enter rate of interest:
5
Interest will be compounded 5 times a year
Enter the choice:
1 .Set the values for savings acc
2. display
3. deposit
4. Interest
5. Withdraw
6. exit
3
Enter the amount to be deposited: 1200
```

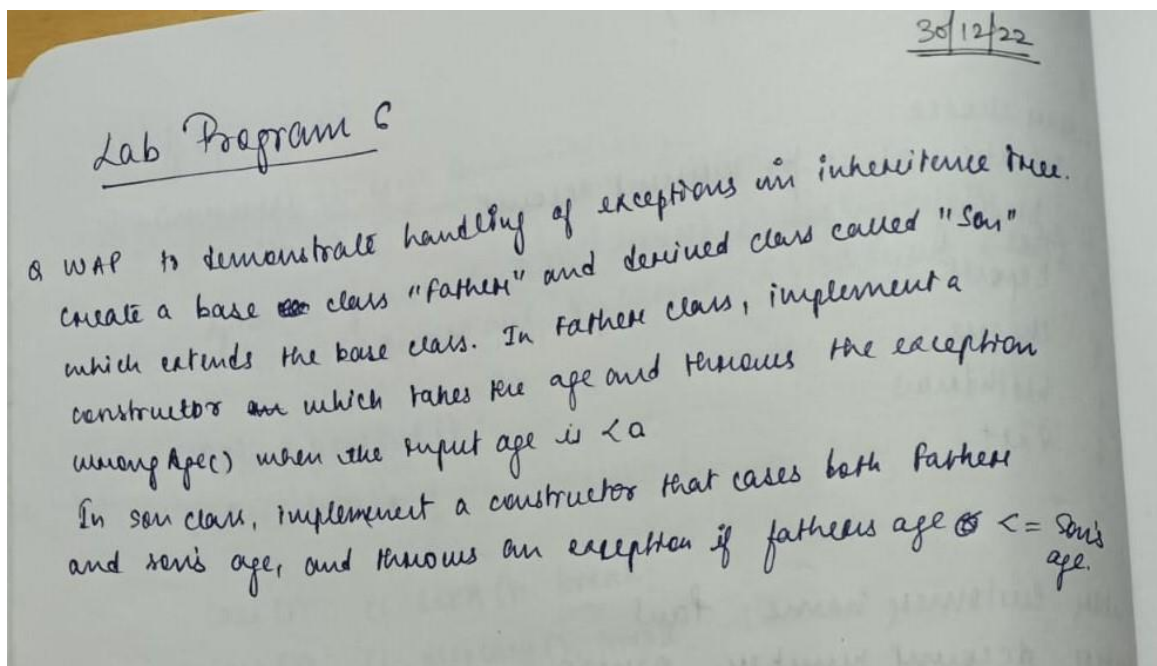


```
C:\Users\anosh\OneDrive\Desktop\java practice>java Bank
1. Savings or 2. Current?
2
Enter the choice:
1.Set the values for current account
2. display
3. deposit
4. transferCheck
5. Withdraw
6. exit
1
Enter customer name: Paul
Enter account number: 024123
Enter bank balance: 8000
Enter the choice:
1.Set the values for current account
2. display
3. deposit
4. transferCheck
5. Withdraw
6. exit
4
Enter the cheque amount: 9000
Rs. 500 penalty imposed...Is it ok to proceed? Enter y for yes and n for no
y
Enter the choice:
1.Set the values for current account
2. display
3. deposit
4. transferCheck
5. Withdraw
6. exit
2
Customer name is: Paul
Customer account type is: Current
Customer account number is: 24123
Current balance is: -1500.0
```

## LAB PROGRAM 6

### QUESTION:

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 takes both father and son's age and throws an exception if son's age is  $\geq$  father's age.



## WRITTEN CODE

```
import java.util.Scanner;

class InvalidAgeException extends Exception
{
    public InvalidAgeException (String str)
    {
        super(str);
    }
}

class Father {
    int age;
    Father () throws InvalidAgeException {
        Scanner ss = new Scanner (System.in);
        System.out.println ("Enter Father's Age:");
        age = ss.nextInt();
        if (age < 0) {
            throw new InvalidAgeException ("age is negative");
        }
        else {
            System.out.println ("Success");
        }
    }
}
```

```
class Son extends Father {
```

```
    int age;
```

```
    Son() throws InvalidAgeException {
```

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

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

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

```
        if (age >= super.age) {
```

```
            throws new InvalidAgeException("Son's Age wrong");
```

```
        }
```

```
    } else {
```

```
        System.out.println("Proper Age");
```

```
    }
```

```
}
```

```
}
```

```
class fsmain {
```

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

```
        try {
```

```
            Son S1 = new Son();
```

```
        }
```

```
        catch (InvalidAgeException ex) {
```

```
            System.out.println(ex);
```

```
        }
```

```
    }
```

```
}
```

## OUTPUT:

① Enter Father's Age

-3

Invalid Age Exception: <sup>age is</sup> Negative

② Enter Father's Age:

20

proper Age

Invalid Age Exception: son's Age is missing

③ Enter Father's Age:

23

proper age

Enter Son's Age

20

proper age.



20/12/2022



# OUTPUT

```
C:\Users\anosh\OneDrive\Desktop\java practice>java fsmain
Enter Father's Age:
20
Proper Age
Enter Son's Age:
12
Proper Age
```

```
C:\Users\anosh\OneDrive\Desktop\java practice>java fsmain
Enter Father's Age:
30
Proper Age
Enter Son's Age:
45
InvalidAgeException: Son's Age is Wrong
```

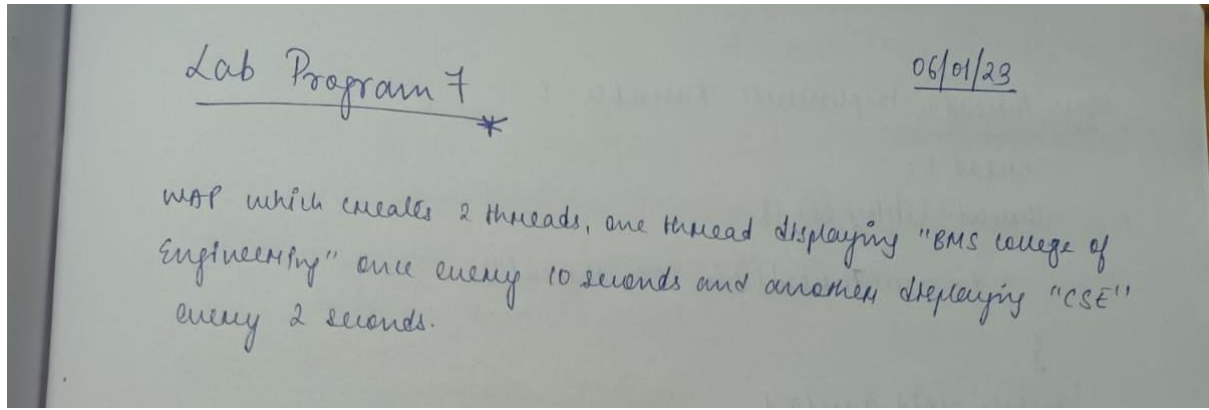
```
C:\Users\anosh\OneDrive\Desktop\java practice>java fsmain
Enter Father's Age:
-23
InvalidAgeException: age is negative
```

```
C:\Users\anosh\OneDrive\Desktop\java practice>java fsmain
Enter Father's Age:
34
Proper Age
Enter Son's Age:
-3
InvalidAgeException: Son's Age is Wrong
```

## LAB PROGRAM 6

### QUESTION:

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.



## WRITTEN CODE

```
class Thread1 implements Runnable {  
    Thread t;  
    Thread1(String ss) { // not using ss here  
        t = new Thread(this, "BMSss Thread");  
        // System.out.println(t);  
    }  
    public void run() {  
        try {  
            for (int i=5; i>0; i--) {  
                System.out.println("BMS college of Engineering");  
                Thread.sleep(10000);  
            }  
        } catch (InterruptedException ie) {  
            System.out.println("Thread interrupted");  
        }  
        // SOP ("BMS Thread exiting")  
    }  
}
```

```
class Thread2 implements Runnable {
```

```
    Thread t;
```

```
    Thread2(String ss) {
```

```
        t = new Thread(this, "CSE Thread");
```

```
    }
```

```
    public void run() {
```

```
        try {
```

```
            for (int i=5; i>0; i--) {
```

```
                System.out.println("CSE");
```

```
                Thread.sleep(2000);
```

```
            }
```

```
        } catch (InterruptedException ie) {
```

```
            System.out.println("Thread interrupted");
```

```
        }
```

```
        // SOP("CSE exiting");
```

```
    }
```

```
}
```

```
class Threadmain {
```

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

```
        Thread t1 = new Thread01("Thread");
```

```
        Thread t2 = new Thread02("Thread");
```

```
        t1.start();
```

```
        t2.start();
```

```
    }
```

```
}
```



OUTPUT:

BMS college of Engineering

CSE

CSE

CSE

CSE

CSE

BMS college of Engineering

BMS college of Engineering

BMS college of Engineering

~~BMS college of Engineering~~

Done

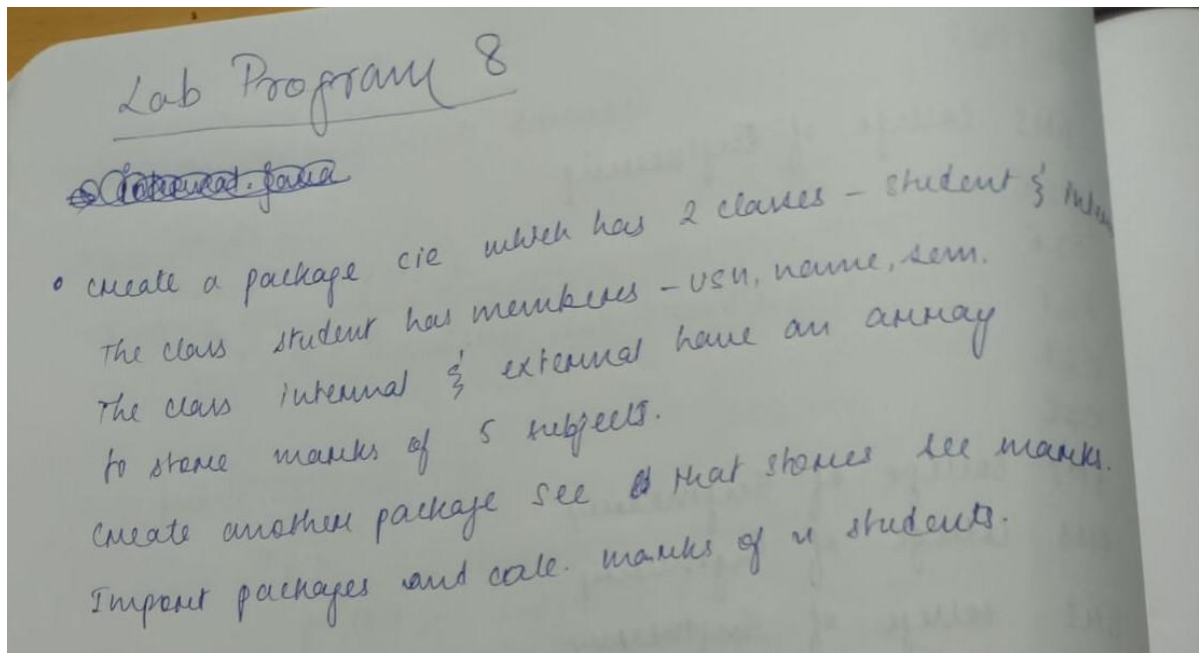
# OUTPUT

```
C:\Users\anosh\OneDrive\Desktop\java practice>javac threadmain.java  
C:\Users\anosh\OneDrive\Desktop\java practice>java threadmain  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
BMS College of Engineering  
BMS College of Engineering  
BMS College of Engineering
```

## LAB PROGRAM 8

### QUESTION 1:

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.



## WRITTEN CODE

### Lab Program 8

~~Internal Java~~

- create a package `cie` which has 2 classes - `student` & `internal`.  
The class `student` has members - `usn`, `name`, `sem`.  
The class `internal` & `external` have an array  
to store marks of 5 subjects.  
Create another package `see` that stores all marks.  
Import packages and call marks of `n` students.

(#1)

- `import java.util.Scanner;`

~~class~~ `public student {`

`String name = new String();`

`String usn = new String();`

`int sem;`

`public student() {`

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

`System.out.println("enter name, usn, sem");`

`name = s.next();`

`usn = s.next();`

`sem = s.nextInt();`

`}`

`}`



```

public class Internal extends Student {
    protected float marks[] = new float[5];
    public Internal() {
        Scanner ss = new Scanner(System.in);
        for (int i = 0; i < 5; i++)
            marks[i] = ss.nextFloat();
    }
}

```

#2

```

package see;
import java.util.Scanner;
import java.util.Scanner;
import see.internal;

public class External extends Internal {
    float marks2[] = new float[5];
    public External() {
        Scanner ss = new Scanner(System.in);
        for (int i = 0; i < 5; i++)
            marks2[i] = ss ss.nextFloat();
    }

    public void calc() {
        for (int i = 0; i < 5; i++) {
            System.out.println(marks[i] + marks2[i]);
        }
    }
}

```

#3

```
import java.util.Scanner;  
import cie.internal  
import cie.external
```

```
class pmain {
```

```
    public static void main (---) {
```

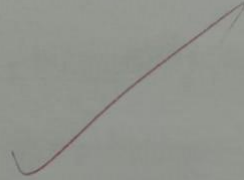
```
        Scanner s = new Scanner (System.in)
```

```
        external b1 = new external();
```

```
        b1.calc();
```

```
    }
```

```
}
```



*Arav*  
13-1-2023

# OUTPUT

```
C:\Users\anosh\OneDrive\Desktop\java practice\New folder>java pmain
Enter number of students:
1
Student 1 details
Enter your name:
Anoshor
Enter your usn:
1BM21CS024
Enter your sem:
3
Enter test marks for cie
Marks for sub1
34
Marks for sub2
43
Marks for sub3
45
Marks for sub4
50
Marks for sub5
40
Enter SEE marks
Enter marks for subject1
99
Enter marks for subject2
92
Enter marks for subject3
78
Enter marks for subject4
83
Enter marks for subject5
80
Student 1 Total Marks
Totat marks for Subject1 83.5
Totat marks for Subject2 89.0
Totat marks for Subject3 84.0
Totat marks for Subject4 91.5
Totat marks for Subject5 80.0
```

## QUESTION 2:

Develop a Generic Class with Two Type Parameters.

## CODE

```
import java.util.Scanner;

class gen1<T1, T2> {
    int x;
    private T1 t1;
    private T2 t2;

    gen1(T1 t1, T2 t2) {
        this.t1=t1;
        this.t2=t2;
    }

    void getvaltype () {
        System.out.println("t1 is of type "+t1.getClass().getName()+" and has value "+t1);
        System.out.println("t2 is of type "+t2.getClass().getName()+" and has value "+t2);
    }
}

class gmain {
    public static void main(String sss[]) {
        gen1<Integer,String> g1=new gen1<Integer,String>(10,"Anoshor");
        g1.getvaltype();
    }
}
```

## OUTPUT:

```
C:\Users\anosh\OneDrive\Desktop\java practice>javac gmain.java

C:\Users\anosh\OneDrive\Desktop\java practice>java gmain
t1 is of type java.lang.Integer and has value 10
t2 is of type java.lang.String and has value Anoshor
```