

# **B.M.S. COLLEGE OF ENGINEERING BENGALURU**

Autonomous Institute, Affiliated to VTU



## **LAB REPORT**

**23CS3PCOOJ**

Submitted in partial fulfilment of requirements for Lab

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By :

**POOJA M**

**(1BM22CS195)**

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024

## INDEX

SL.NO	Title	Date
1	Complete scanned Observation Book	12/12/2023-20/02/24
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

## Lab prgm-1

### Q Quadratic

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

```
class Quadratic {
```

```
{
```

```
    int a, b, c;
```

```
    double r1, r2, d;
```

```
    void getd()
```

```
{
```

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

```
    System.out.println("Enter the co-efficients of a, b, c");
```

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

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

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

```
g
```

```
    void compute()
```

```
{
```

```
    while (a == 0)
```

```
{
```

```
    System.out.println("Not a quadratic equation");
```

```
    System.out.println("Enter a non zero value c for a");
```

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

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

```
g
```

```
    d = b * b - 4 * a * c;
```

```
    if (d == 0)
```

```
{
```

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

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

```
        System.out.println("Root 1 = Root 2");
```

```
g
```

else if ( $d > 0$ )  
 {

$$r_1 = ((-b) + (\text{Math.sqrt}(d))) / (\text{double}(2 * a));$$

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

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

System.out.println ("Root 1 = " + r1 + " Root 2 = "  
+ r2);

}

else if ( $d < 0$ )  
 {

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

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

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

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

System.out.println ("Root 1 = " + r1 + " - " + "i" + r2);

}

}

class QuadraticMain

{

public static void main (String args{})

{

Quadratic q = new Quadratic();

q.getd();

q.compute();

}

Output .

1) Enter the coefficients of a, b, c

23 44 55

+ r1); Roots are imaginary

$$\text{Root}_1 = 0.0 + i1.2150598793462712$$
$$\text{Root}_2 = 0.0 - i1.2150598793462712$$

2. Enter the coefficients of  $a, b, c$ .

2 4 2

Roots are real and equal.

$$\text{Root}_1 = \text{Root}_2 = -1.0$$

3. Enter the coefficients of  $a, b, c$ .

2 6 2

Roots are real and distinct.

$$\text{Root}_1 = -0.381966$$

$$\text{Root}_2 = -2.61803$$

Enter the coefficients of  $a, b, c$

0 0 0

~~Enter~~ not a quadratic equation.

Enter a non-zero value for  $a$ ;

Enter the coefficients equation

0 2 3.

Enter a quadratic equation.

Enter a non zero value for  $a$

## LAB prog-2.

```

import java.util.Scanner;
class Subject {
    int subjectMarks, credits, grade;
}
class Student {
    String name;
    String usn;
    double CGPA;
    Scanner s;
    Subject subjects[];
}

```

## Student()

{

int i;

subjects = new Subject[9];

for (i=0; i&lt;8; i++)

subjects[i] = new Subject();

s = new Scanner(System.in);

g

## public void getStudentDetails()

System.out.println("Enter student name:");
name = s.nextLine();

System.out.println("Enter student USN:");
usn = s.nextLine();

## public void getMarks()

int i;

for (i=0; i&lt;8; i++)

System.out.println("Enter marks of subject " + (i+1) + ":");
subjectMarks[i].subjectMarks = s.nextInt();

if (subjectMarks[i].subjectMarks &gt;= 40 &amp; &amp; subjectMarks[i].subjectMarks &lt;= 100)

subjectMarks[i].grade = calculateGrade

Subject[i].grade = calculateGrade

(subject[i].subjectMarks);  
else {

System.out.println ("Invalid Marks  
Marks should be between 40 and  
100");

System.out.print ("Enter credits  
subjects[i].credits = s.nextInt());  
q  
q

```
public int calculateGrade (int marks)  
{  
    if (marks >= 90)  
        return 10;  
    else if (marks >= 70 && marks <= 80)  
        return 9;  
    else if (marks >= 60 && marks <= 70)  
        return 8;  
    else if (marks >= 50 && marks <= 60)  
        return 7;  
    else  
        return 6;
```

```
for (i = 0; i < 8; i++)  
    totalScore += subjects[i].grade *  
        subjects[i].credits;  
    totalCred += subjects[i].credits;
```

float avg = (double) totalScore / (double)  
totalCred;

public class student

public static void main (String args)

{  
student s1 = new student();

s1.get student Details();

s1.get Marks();

s1.compute SGPA();

System.out.println ("student name:"  
+ s1.name);

System.out.println ("student USN : " + s1.usn);

System.out.println ("student SGPA : " +  
s1.SGPA);  
}

### Output

Enter student name:  
manya.

Enter student USN:

109883

Enter marks of subjects 1:  
88

Enter credits:

8

Enter marks of subjects 2:  
78

Enter credits:

7

Enter marks of subjects 3:  
89.

Enter credits:

9.

Enter marks of subjects 4:  
78

Enter credits:

7

Enter marks of subjects:

88

Enter credits:

8

Enter marks of subjects:

62

Enter credits:

7

Enter marks of subjects:

8.

Enter credits:

7

Enter marks of subjects:

8.

Enter credits:

7

Student name: mary a.

Student USN: 109283.

Student sgpa: 7.62.

3

LAB Program -3.

```
import java.util.Scanner;  
class Book {  
    String name;  
    String author;  
    int price;  
    int numPages;  
  
    Book(string name, string author,  
          int price, int numPages)  
    {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.numPages = numPages;  
    }  
  
    public string toString()  
    {  
        string bookDetails = "Book name : "  
            + this.name + "\n" + "Author  
name : " + this.author + "\n"  
            + "Price : " + this.price + "  
" + "Number of pages : " + this  
numPages + "\n";  
        return bookDetails;  
    }  
  
    public class Main {  
        public static void main(string args)  
        {  
            Scanner s = new Scanner(System.in);  
            System.out.print("Enter the  
number of books : ");  
            int n = s.nextInt();  
        }  
    }  
}
```

BOOK [ ] books = new BOOK [n];

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

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

System.out.println("Name : ");  
String name = s.nextLine();

System.out.println("Author : ");  
String author = s.nextLine();

System.out.println("Price : ");  
int price = s.nextInt();

System.out.println("Number of  
pages : ");

int numPage = s.nextInt();

books[i] = new Book(name, author,  
price, numPage);

System.out.println("In Details");

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

System.out.println("Book"  
+ (i + 1) + ": " + books[i].strings);

## Output

Enter the number of books : 2

Enter details for Book 1

Name : Grids

Author : Jamie

Price : 300

Number of pages : 670

Enter details for Book 2.

Name : CPVGM

Author : Rema

Price : 679

Number of pages : 700

Details of the books :

Book 1 :

Book name : Grade

Author name : Jamie

Price : 300

Number of pages : 670

Book 2 :

Book name : CPVGM

Author name : Rema

Price : 679

Number of pages : 700

23  
24  
25  
26

19(1);

## LAB program - 4.

```
import java.util.Scanner;
class InputScanner {
    Scanner s = new Scanner (System.in);
    double getInput (String prompt) {
        System.out.println (prompt);
        return s.nextDouble ();
    }
    abstract class Shape extends
        InputScanner {
        double side1, side2;
        abstract void area();
    }
}
```

```
class Rectangle extends Shape {
    Rectangle () {
        side1 = getInput ("Enter length
            of rectangle:");
        side2 = getInput ("Enter breadth
            of rectangle:");
    }
    void area () {
        double area = side1 * side2;
        System.out.println ("Area of the
            rectangle = " + area);
    }
}
```

```
class triangle extends Shape {
    triangle () {
        side1 = getInput ("Enter
            base of the triangle:");
        side2 = getInput ("Enter
            height of the triangle");
    }
    void area () {
        double area = side1 * side2
            / 2;
    }
}
```

double area = side1 \* side2 / 2;

~~System.out.println ("Area of the triangle = " + area); } }  
class circle extends shaped circle { }~~

~~circle r = getinput ("Enter the radius of the circle:"); } }  
void area() { }~~

~~double area = Math.PI \* r \* r; } }  
System.out.println ("Area of the circle = " + area); } }~~

~~class Main { }~~

~~public static void main (String args[]) { }~~

~~Rectangle rectangle = new Rectangle();  
triangle triangle = new triangle();  
circle circle = new circle(); }~~

~~rectangle.area(); }~~

~~triangle.area(); }~~

~~circle.area(); }~~

Output:-

Enter length of rectangle:

20

Enter breadth of rectangle:

40

Enter base of the triangle:

6

Enter height of the triangle:

8

Enter the radius of the circle:

5

area of Rectangle = 800.0

~~the~~ ~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

~~the~~ ~~the~~ ~~the~~

class CurrentAccount extends Account {  
 double minBalance;  
 double serviceCharge;

public CurrentAccount (String customerName,  
 long accountNumber, double balance,  
 double minBalance, double serviceCharge);

super (customerName, accountNumber,  
 accountType = "current", balance);  
 this.minBalance = minBalance;  
 this.serviceCharge = serviceCharge;

public void checkMinBalance () {

if (balance < minBalance) {

balance -= serviceCharge;

System.out.println ("Minimum  
 balance not maintained. Service  
 charge of INR" + serviceCharge + "imposed");

displayBalance();

}

public void withdraw (double amount) {

if (amount > balance) {

System.out.println ("Insufficient  
 funds. withdrawal failed.");

else {

balance -= amount;

System.out.println ("withdrawal  
 of INR" + amount + "successful.");

updated balance: INR" + balance);

checkMinBalance();

}

g

g

class SavingsAccount extends Account {  
 double interestRate;}

public SavingsAccount (String customerName,  
 long accountNumber, double balance,  
 double interestRate) {  
 super (customerName, accountNumber,  
 "Savings", balance);  
 this.interestRate = interestRate;

q.

public void computeInterest() {  
 double interest = balance \* (interest  
 Rate / 100);

balance += interest;

System.out.println ("Interest  
 computed and deposited: INR"  
 + interest);

displayBalance();

q.

public void withdraw (double amount)

if (amount > balance) {

System.out.println ("Insufficient  
 funds, withdrawal failed");

else {

balance -= amount;

System.out.println ("withdrawal  
 of INR" + amount + "successful.

Update balance: INR" + balance);

q.

q.

q.

public class Bank {

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

```
Scanner scanner = new Scanner (System.in);
Cust Account currentAccount = new CustAccount
    (customerName : "John Doe",
     accountNumber : 123456789, balance : 1000,
     minBalance : 500, serviceCharge : 10);
```

```
sav Account savings Account = new
    sav Account (CustomerName : "Jacob",
     accountNumber : 987654321, balance :
     2000, interestRate : 5);
```

```
int choice;
```

```
do
```

```
System.out.println ("In Select an
option :");
```

```
System.out.println ("1. Deposit");
```

```
System.out.println ("2. Display Balance");
```

```
System.out.println ("3. Compute Interest
(savings Account only));
```

```
System.out.println ("4. withdraw");
```

```
System.out.println ("5. Exit");
```

```
System.out.println ("Enter your choice");
choice = scanner.nextInt();
```

```
switch (choice) {
```

```
case 1:
```

```
System.out.println ("Enter
amount to deposit :").
```

```
double depositAmount = scanner.
    nextDouble();
```

```
System.out.print ("Select account
```

```

    (1. Current , 2. Savings) ;") ;
    int amountType = scanner.nextInt();
    if (amountType == 1) {
        currentAmount.deposit(depositAmount);
    }
    else if (amountType == 2) {
        savingsAmount = deposit(depositAmount);
    }
    else {
        System.out.println("Invalid
                            amount type.");
        break;
    }
    case 2:
        System.out.print("Enter amount to
                        deposit : ");
        System.out.println("Select account
                            (1. Current, 2. Savings)");
        int autype = scanner.nextInt();
        if (autype == 1) {
            currentAmount.displayBalance();
        }
        else if (autype == 2) {
            savingsAmount.displayBalance();
        }
        else {
            System.out.println("Invalid
                                option for current account.");
        }
    }
    case 3:
        System.out.print("Enter amount
                        to withdraw : ");
        withdrawAmount = scanner.nextInt();
    }
}

```

```
System.out.print("Select an option (1. deposit  
, 2. savings):");
```

```
int autyp = scanner.nextInt();  
if (autyp == 1) {
```

```
currentAmount.withdraw(withdrawalAmount);
```

```
} else if (autyp == 2) {
```

```
savingsAmount.withdraw(withdrawalAmount);
```

```
} else
```

```
System.out.println("Invalid amount  
type");
```

```
by exit;
```

```
case 5:
```

```
System.out.println("Invalid choice,  
please enter valid option");
```

```
} while (choice != 5);
```

```
scanner.close();
```

```
} // end of class
```

Output:

Select an option:

1. Deposit

2. Display Balance.

3. Compute Interest (savings Amount  
only).

4. withdraw.

5. Exit.

Enter your choice:

Enter amount of deposit: 500000

Select amount (1. current, 2. saving): 1

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

Deposit of INR 500000, successful.  
Updated Balance : INR 501000.0

Select an option :

1. Deposit.
2. Display Balance.
3. Compute Interest.
4. Withdraw
5. Exit.

Enter your choice : 2.

Select account (1. Current, 2. Savings) ?  
Amount Balance : INR 501000.0

Select an option :

- 1.
- 2.
- 3.
- 4.
- 5.

Enter your choice : 3.

Interest computed and deposited : INR 100.00  
Amount Balance : INR 501100.0

Select an option .

- 1
- 2
- 3
- 4
- 5.

Enter amount to withdraw : 60000.0

Select account : 1.  
Withdrawal of INR 60000.0

Select an option:

- 1
- 2
- 3
- 4
- 5

Enter your choice : 2.

Select account : 1.

Amount Balance : INR 44,000.0.

Select an option:

- 1
- 2
- 3
- 4
- 5

Enter your choice : 1.

Enter amount to deposit : 400000

Select amount : 2

Deposit of INR 400000 successful.

Updated balance.

Select an option:

- 1.
- 2
- 3
- 4
- 5

Enter your choice : 3.

Entered computed and deposited :

INR 200005.0.

Amount Balance : INR 420005.0

Select an option:-

1

2

3

4

5

Enter your choice : 2.

Select account - 2.

Amount Balance: INR 4412315

Select an option:-

1

2

3

4

5

Enter your choice : 4

Enter amount to withdraw : 200000

Select account : 2.

withdrawal of INR 200000.0 successful

Select an option:

~~Enter your choice : 5~~

~~Exiting the program. Thank you!~~

## LAB - 6.

1. Hello, World!  
Hello  
Hello  
Java.

2. String Length : 13  
String Literal 1 : Java.  
String Literal 2 : Java.  
are string string Equal ? true.  
concatenated string - Program

3. Dimensions are 10.0 by 14.0 by  
12.0 Box is Dimensional size 10.0  
by 14.0 by 12.0

4. Extracted substring - Bmsle

5. Byte Array:

Byte: 72 101 108 111 44 32 87 111 114  
108 100 33.

- char Array:

char : E d j a v a A d d D r i n k i n g m i l k s h a k e

6. Using equals(): Bmsle equals Bmle  
→ true

Using equals(): Bmle equals College →  
False

Using equals(): Bmsle equals BMsle →  
false

Using equalsIgnoreCase(): Bmle  
equals Ignore  
case BMSLE →

7. Substring is matched.

8. Starts with "Hello": true.  
Starts with "PROGRAM": False.

9. Ends with "Buddy!": true.  
Ends with "PROGRAM": false.

10. Hello equals hello → true  
Hello == hello → false.

11. sorted words:

apple	gun	man	star	gatch
ball	gen	nct	tric	zcc
bent	ice		orange	umbrella
dog	jug	paratran		
cnt	kitt	greenwatch		
frcc	lift	ring	xmat	

12. Sort id Numbers: 11023456789.

13. Original string: This was was a test  
This was was, too  
After Replacement: This is a test  
This is, too.

14. concatenated string: holloworld.

15. Original string: welcome to college.  
Modified string: welcome to compmoc

16. Original string: 'Hello jnckh'  
Trimmed string: 'Hello Friends'

12. Enter details for student 1:

Registration Number: 101

Full Name: pooya.

Semester: 3

CGPA: 8.9.

Enter details for student 2:

Registration Number: 102

Full name: prabhanjan

Semester: 3

CGPA: 9.7.

Enter details for student 3:

Registration Number: 103

Full name: Vinutha

Semester: 3

CGPA: 9.8

18. After setLength(5); Hello

char At(1): after setChar(1, 'a'):

Helloat char[0, 5, Char Array, 0]: Hello

After append: Hello World : After

insert(6, " Java "): Hello,

JavaWorld !

After reverse: olleWorl d, ollcN.

After delete(3,11): lly, ollen.

After deleteChar(11): lly, olen.

After replace(7,12, "World"):

Hello, World Substring (7, 12):

: World.

19. Eagle: Eagle flies high in the sky,

Eagle screeches loudly.

Hawk:

~~16/10/2023~~ Hawk soars gracefully through the air.  
Hawk emits a piercing cry.

90. Circle :

Area :  $\pi \approx 3.141592653589793$

Perimeter :  $2\pi r \approx 6.283185307179586$

~~Triangle :~~

~~Area : 6.0~~

~~Perimeter : 12.0~~

~~Tri  
16/01/24~~

## Lab

23/07/24

Packages

code :-

package CSE;

public class Internals {

private int[] internalMarks = new int[6];

public Internals() {

}

public void setInternalMarks(int[]

internalMarks) {

this.internalMarks = internalMarks;

}

public int[] getInternalMarks() {

return internalMarks;

}

}

package CSE;

public class Student {

public String usn;

public String name;

public int sem;

public Student() {

this("", "", 0);

}

public Student(String usn, String name,  
int sem) {

this.usn = usn;

this.name = name;

this.sem = sem;

}

public void setUsn(String usn) {

this.usn = usn;

}

public void setName (String name) {  
    this.name = name;

{}

public void setSem (int sem) {  
    this.sem = sem;

{}

public String getUsn () {  
    return usn;

{}

public String getName () {  
    return name;

{}

public int getSem () {

    return sem;

{}

package SEE;

import EIE.Student;

public class External extends Student

    public int [] seeMarks = new int [

        public External () {

            this ( "", "", 0, new int [5] );

        public External (String usn, String name,

            int sem, int [] seeMarks) {

            super (usn, name, sem);

            this . seeMarks = seeMarks;

{}

        public void setseeMarks (int [] seeMarks)

            this . seeMarks = seeMarks;

{}

public int[] getSCEMarks() {  
 "Enter n SCE Marks";

g.  
g.

```
import CIE.Student;
import CIE.Internal;
import SGE.External;
import java.util.Scanner;
```

public class FinalMarks {

public static void main (String [] args)

{  
 Scanner scanner = new Scanner (System.in);

~~System.out.print ("Enter the number  
 of students") -~~

int n = scanner.nextInt();

Student [] students = new Student [n];  
 };

Internal [] internal = new Internal [n];

External [] external = new External [n];

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

Student student [] = new Student ();

System.out.print ("Enter USN  
 for student");

+ (i+1) + ":" );

student.setUrn (scanner.nextInt());

System.out.print ("Enter name");

for (Student student [] = (i+1) + ":" );

student.setName (scanner.next());

System.out.print ("Enter semester for  
 student" + (i+1) + ":" );

`students[i].getSem(scanner.nextInt())`

`internals[i] = new Internal()`

`internals[i].setInternalMarks(scanner.nextInt(),  
marksWithValidation("internal", i))`

`externals[i] = new External(students[i],  
new int[5])`

`externals[i].setExternalMarks(scanner.nextInt(),  
validation("external", i, scanner, 0))`

`int[] finalMarks = new int[5]`

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

`finalMarks[j] = internals[i].getInternals()  
getSEEMarks(i) / 2;`

`System.out.println("Student" + i +)`

`finalMarks[0] + ". " + finalMarks[1] +  
" " + finalMarks[2] + ", " + finalMarks[3] + "  
finalMarks[4]);`

`scanner.close();`

`g`

`private static int[] inputMarksWithValidation  
(String type, int studentIndex, scanner,  
int min, int max) {`

`int[] marks = new int[5];`

`System.out.println("Enter " + type + "  
student" + (studentIndex + 1) + ":");`

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

`int mark;`

`do {`

`System.out.print("Subject" + (i + 1) + ":");`

`mark[i] = mark;`

`g`

`return no mark;`  
`g g.`

`mark, 0, 30));`

Output :-

Enter the number of  
Students = 2

`getUvn()`

Enter USN for Student 1:  
101

`stt[i].getSem(),`

Enter name for Student 1:  
Nitisha.

~~Enter internal marks for  
Student 1:~~

Subject 1 : 44

Subject 2 : 41

Subject 3 : 43

Subject 4 : 42

Subject 5 : 44.

`+ "FinalMarks") ;`

Enter external result for  
Student 1 :

Subject 1 : 89

Subject 2 : 88

Subject 3 : 87

Subject 4 : 88

Subject 5 : 78.

Student 1 Final Marks : 89, 88  
86, 88, 80.

Enter USN for Student 2:  
102.

Enter name for Student  
2 : Pooja

Enter semester for  
Student 2 : 2.

Enter internal marks for student 2.

Subject 1 : 43

Subject 2 : 44

Subject 3 : 39

Subject 4 : 44

Subject 5 : 44.

Enter external marks for student 3 :

Subject 1 : 98

Subject 2 : 88

Subject 3 : 89

Subject 4 : 89

Subject 5 : 89.

Student 2 Final Marks : 93, 88, 81, 88, 88

24  
01-24  
24

## LAB-7.

1) Father Son's age.

code:-  
import java.util.Scanner;  
class WrongAge extends Exception  
{  
 WrongAge (String error);  
 System.out.println(error);  
}  
q  
q

class Father {  
 int age;

Father (int age) throws WrongAge {  
 if (age <= 0)  
 throw new WrongAge ("Father's age can't be negative");  
 this.age = age;

q  
q

class Son extends Father {  
 int age;

Son (int age, int s-age) throws  
WrongAge {  
 super (age);  
 if (s-age <= age)

throw new WrongAge ("Son's age can't be greater than Father's age");  
 this.s-age = s-age;

q  
q

class Lab7 {

public static void main (String args[]) {  
 Scanner sc = new Scanner (System.in);  
 try {  
 System.out.println ("Enter the Father's Age:

```

int f_age = sc.nextInt();
System.out.println("Enter the son's age:");
int s_age = sc.nextInt();
Son a = new Son(f_age, s_age);
System.out.println("Father's age : " + f_age);
System.out.println("Son's age : " + s_age);
}
catch (WrongAgeException e) {
System.out.println("Wrong Age entered");
}
catch (Exception e) {
System.out.println("Unexpected error");
}
}

```

Output:-

Enter the Father's Age:  
48

Enter the son's Age:  
19

Father's age : 48  
Son's age : 19.

Enter the Father's Age:  
18

Enter the son's Age:  
25

Son's age can not be greater than  
Father's Age.

Wrong age entered.

Enter the Father's age:

-20

Enter the Son's age:

23

~~30.08.2014~~ Father's age cannot be negative

Wrong Age entered.

6 | 2 | 24

## Week- 8

## Creating two threads:

word e:-

class Display Thread extends Thread {

~~private String message;~~

private int interval;

private boolean running = true;

public Display Thread (String message,  
int interval) {

this message = message;

this interval is internal;

giant, inland

public void run();

~~System.out.println(museg);~~

try d.

Thirst, sleep (interval)

g) catch (Interrupted Exception)  
e) d.

e. printstackTrace();

۹۳

9

public void stopThread() {

3 running = false;

g

public class Thread2

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

Display Thread 6ms Thread = new Display

Third ("BM's College of  
Engineering", 1900).

Display Thread < Thread = new DisplayThread ("CSE", 2000);

```
bmdThread.start();  
use Thread.start();  
. System.out.println ("Press Enter to  
stop the threads...");  
try {
```

```
    System.in.read();  
} catch (Exception e) {  
    e.printStackTrace();
```

```
    bmdThread.stopThread();  
    use Thread.stopThread();
```

Output:

BMS College of Engineering

CSB

CSB

CSE

CSB

CSE

BMS College of Engineering.

ESE

CSE

CSE

CSE

BMS College of Engineering.

ESE

CSE

CSE

6/2/24

## Week -10 , JPL + Deadlock .

```

code:-                                b(1) q(1) p(1)
class Q L.                         i(1) j(1) k(1)
int n = 0;
boolean valueset = false;
synchronized int get() {
    while (c ! valueset)
        try {
            wait();
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
    System.out.println("Get : " + n);
    valueset = true;
    notify();
    return n;
}

```

Synchronized void put(int n) {

while (valueset);

try {

wait();

} catch (InterruptedException e) {

System.out.println("Put : " + n);

notify();

} }

class Producer implements Runnable.

Q q;

Producer (Q q) {

this.q = q;

new Thread (this, "producer").start();

}

public void run() {

int i = 0;

```
while (i < 15) {
    q.put (i++);
}
q
q
q

class Consumer implements Runnable {
    Queue q;
    Consumer(Q q) {
        this.q = q;
    }
    new Thread (this, "Consumer").start();
}

public void run() {
    int i=0;
    while (i < 10) {
        int r = q.get();
        i++;
    }
}
q
q
q
```

class PCFixed.

```
public static void main (String arg) {
    Queue q = new Queue();
    new producer (q);
    new consumer (q);
    System.out.println ("Press control - stop.");
}
q
q
```

st put:-

ut : 0

ut : 1

ut : 1

ut : 2

Got: 2	Got: 4	Got: 6	Got: 8
Put: 3	Put: 5	Put: 7	Put: 9
Got: 3	Got: 5	Got: 7	Got: 9
Put: 4	Put: 6	Put: 8	Put: 10
			Got: 10

## ~~to~~ Deadlock.

06/08/20

Dead lock.

code:-

class A {

synchronized void foo (B b) {

String name = Thread.currentThread().

getName();

System.out.println(name + " entered A");

try {

Thread.sleep(1000);

catch (Exception e) {

System.out.println("A Interrupted");

            System.out.println(name + " trying local  
                break");

break;

void last() {

System.out.println("Inside A.last");

}

class B {

synchronized void bar(A a) {

String name = Thread.currentThread().

getName();

        System.out.println(name + " entered B  
                bar");

try {

Thread.sleep(1000);

catch (Exception e) {

System.out.println ("13 Intercepted")  
g.

System.out.println (name + " trying  
to call A.last()");  
a.last(); g.

void last () {

System.out.println ("Inside A.last()  
g. g.

class Deadlock implements Runnable

{

A a = new A();

B b = new B();

Deadlock () {

Thread.currentThread().sleep(  
("Main Thread"));

Thread t = new Thread (this,  
"Racing Thread");

t.start();

t.join();

System.out.println ("Balk  
in main thread");

Output :-

Main Thread entered A.foo

Racing Thread entered B.bar

Main thread trying to call B.last().  
Inside A.last.

BALK in main thread

Racing Thread trying to call A.last().  
tryed e. A.last.

BALK in other thread

Create an user interface to  
perform integer division of  
classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

## LAB - 9.

### Swing Demo.

Code:-

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;  
class SwingDemo2  
SwingDemo2()  
JFrame fm = new JFrame("Divide");  
fm.setSize(275, 150);  
fm.setLayout(new FlowLayout());  
fm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
JLabel glab = new JLabel("Enter the  
divident and divisor :");
```

```
JTextField a JTextField a = new JTextField();  
JTextField b JTextField b = new JTextField();
```

```
JButton button = new JButton("Calculate");
```

```
JLabel er = new JLabel();  
JLabel ab = new JLabel();  
JLabel blab = new JLabel();  
JLabel anlab = new JLabel();
```

```
fm.add(er);  
fm.add(ab);  
fm.add(blab);  
fm.add(anlab);  
fm.add(a);  
fm.add(b);  
fm.add(button);
```

Action Listener 1 = new Action;  
public void actionPerformed (ActionEvent e)  
{  
System.out.println ("Action event  
is triggered");  
}

`ajt.add ActionListener (l);`  
`bjt.add ActionListener (l);`

button . add ActionListener

public void actionPerformed(ActionEvent e) {  
try {

`int a = Integer.parseInt(str);`

```
int b = Integer.parseInt(  
    int a;
```

int ans = a/b;

alab.setTex + ("ln A = " + a);

play.setText ("In B = "+b);

anslab.setText ("In Ans = " +

catch (NumberFormatException e) {  
 // handle exception  
}

~~alab.setText(" ")~~

6(a). setText (" " )

antlab.setText(" ") .

err.SetText ("Enter only Int")

```
catch (ArithmeticalException e) {  
    System.out.println("An error occurred");  
}
```

```
abab.setTxt(" ..")
```

6. lab. set Text (" ")

~~arr1ab.setText(" ")~~

~~cytText ("18 should~~

- A point is shown below.

a

9

gj;

$i \mapsto m$

150

```

next();
Event evt) {
    JTextField id);
    public static void main
        (String args[]) {
        SwingUtilities.invokeLater
            (new Runnable () {
                public void run () {
                    new swing Demo();
                }
            });
    }
}

```

`.getText();`      Output -  
`t.getText();`      Divider App  
 Enter the dividend and  
 divisor:

200

10

Calculate

$A = 200 \quad B = 10 \quad Quotient = 20$

20.02.94

Report.

on

AWT Functions -

JFrame: It is a class in Java that is part of the Swing library, which is used for creating graphical user interfaces in Java applications.

setSize(): setSize() is a method which is used with components such as JFrame, JPanel etc to set size.

setLayout(): It is a method which is used to set the layout manager

for a container, such as JPanel,  
any other container component.

setDefaultCloseOperation(): It is a  
method which is used to specify  
default close operation for the

JLabel: It is a class which is used  
to display non-editable text area  
on a GUI.

JTextField: It is a class that provides  
a text input field in a GUI.

AddActionListener: add ActionListener is a  
method that is used to register  
an Action Listener for a component,  
typically for a button or any other  
interactive component that generates  
action events.

setText: It is method used to set the  
content of text-based component.

Ans 02/24  
go

## LAB 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 discriminant  $b^2 - 4ac$  is negative, display a message stating that there are no real solutions

Code:-

```
import java.util.Scanner;

class Quadratic

{
    int a, b, c;
    double r1, r2, d;

    void getd()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the coefficients of a,b,c");
        a = s.nextInt();
        b = s.nextInt();
        c = s.nextInt();
    }

    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic equation");
            System.out.println("Enter a non zero value for a:");
            Scanner s = new Scanner(System.in);
            a = s.nextInt();
        }

        d = b*b-4*a*c;
        if(d==0)
        {
```

```

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

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

System.out.println("Root1 = " + r1);
}

else if(d>0)

{
    r1 = ((-b)+(Math.sqrt(d)))/(double)(2*a);
    r2 = ((-b)-(Math.sqrt(d)))/(double)(2*a);

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

    System.out.println("Root1 = " + r1 + " Root2 = " + r2);
}

else if(d<0)

{
    System.out.println("Roots are imaginary");

    r1 = (-b)/(2*a);
    r2 = Math.sqrt(-d)/(2*a);

    System.out.println("Root1 = " + r1 + " + i" + r2);
    System.out.println("Root1 = " + r1 + " - i" + r2);
}
}
}

```

```

class QuadraticMain

{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();

        q.compute();
    }
}

```

```
}
```

Output:-

```
PS C:\Users\ADMIN\Desktop\1BM22CS193> javac QuadraticMain.java
PS C:\Users\ADMIN\Desktop\1BM22CS193> java QuadraticMain
enter the coefficients of a,b,c
1 2 1
Roots are real and equal
Root1 = Root2 = -1.0
PS C:\Users\ADMIN\Desktop\1BM22CS193> javac QuadraticMain.java
PS C:\Users\ADMIN\Desktop\1BM22CS193> java QuadraticMain
enter the coefficients of a,b,c
2 6 2
Roots are real and distinct
Root1 = -0.3819686112581051 Root2 = -2.618833988749895
PS C:\Users\ADMIN\Desktop\1BM22CS193> javac QuadraticMain.java
PS C:\Users\ADMIN\Desktop\1BM22CS193> java QuadraticMain
enter the coefficients of a,b,c
1 1 1
Roots are imaginary
Root1 = 0.0 + 10.8666254837804386j
Root1 = 0.0 - 10.8666254837804386j
PS C:\Users\ADMIN\Desktop\1BM22CS193> javac QuadraticMain.java
PS C:\Users\ADMIN\Desktop\1BM22CS193> java QuadraticMain
enter the coefficients of a,b,c
0 1 5
Not a quadratic equation
Enter a non zero value for a:
2
Roots are imaginary
Root1 = 0.0 + 11.5612494995995996j
Root1 = 0.0 - 11.5612494995995996j
```

## 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

Code:-

```
import java.util.Scanner;

class subject{ int subjectMarks, credits, grade; }

class Student {

    String name;

    public int calculateGrade(int marks){

        if (marks>=90)

            return 10;

        else if(marks>=70&&marks<=80)

            return 9;

        else if(marks>=60&&marks<=70)

            return 8;

        else if(marks>=50&&marks<=60)

            return 7;

        else
```

```
return 6;
}

public void computeSGPA() {
    int totalscore = 0;
    int totalcred = 0;
    for (int i = 0; i < 8; i++) {
        totalscore += subjects[i].grade * subjects[i].credits;
        totalcred += subjects[i].credits;
    }
    SGPA = (double) totalscore / (double) totalcred;
}
}

class Stud{
    public static void main(String args[]){
        Student s1=new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        System.out.println("Student name:"+s1.name);
        System.out.println("Student usn:"+s1.usn);
        8
        System.out.println("Student sgpa:"+s1.SGPA);}
}
```

## Output:-

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin\Desktop\18M22CS195> javac Stud.java
PS C:\Users\Admin\Desktop\18M22CS195> java Stud
Enter student name:
manyा
Enter Student USN:
109283
Enter marks of subject1:
88
enter credits:
8
Enter marks of subject2:
78
enter credits:
7
Enter marks of subject3:
89
enter credits:
9
Enter marks of subject4:
78
enter credits:
7
Enter marks of subject5:
88
enter credits:
8
Enter marks of subject6:
87
enter credits:
8
Enter marks of subject7:
78
enter credits:
7
Enter marks of subject8:
78
enter credits:
7
Student name:manyा
Student usn:109283
Student sgpa:7.62711864067797
PS C:\Users\Admin\Desktop\18M22CS195> |
```

## 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

### Code:-

```
import java.util.Scanner;

class Book {
    String name;
    String author;
    int price;
    int numPages;

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

    public String toString() {
        String bookDetails = "Book name: " + this.name + "\n"
            + "Author name: " + this.author + "\n"
            + "Price: " + this.price + "\n"
```

```

        + "Number of pages: " + this.numPages + "\n";
    return bookDetails;
}
}

public class Main {
    public static void main(String args[]) {
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of books: ");
        int n = s.nextInt();

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1));
            System.out.println("Name: ");
            String name = s.next();
            System.out.println("Author: ");
            String author = s.next();
            System.out.println("Price: ");
            int price = s.nextInt();
            System.out.println("Number of pages: ");
            int numPages = s.nextInt();

            // Create a new Book object with the entered details
            books[i] = new Book(name, author, price, numPages);
        }

        System.out.println("\nDetails of the books:");
        for (int i = 0; i < n; i++) {
            System.out.println("Book " + (i + 1) + ":" + books[i].toString());
        }
    }
}

```

Output:-

```
C:\Users\Languagr lab 22\Desktop\NEWWW>java Main
Enter the number of books: 2
Enter details for Book 1
Name: Grads
Author: Jamie
Price: 300
Number of pages: 670
Enter details for Book 2
Name: cprgm
Author: reema
Price: 679
Number of pages: 700

Details of the books:
Book 1:
Book name:Grads
Author name:Jamie
Price:300
Number of pages: 670

Book 2:
Book name:cprgm
Author name:reema
Price:679
Number of pages: 700
```

#### 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 classShape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape

Code:-

```
import java.util.Scanner;
class InputScanner{
Scanner s= new Scanner(System.in);
double getInput(String prompt){
System.out.println(prompt);
return s.nextDouble();}
abstract class Shape extends InputScanner{
double side1, side2;
abstract void area();}

class Rectangle extends Shape{
Rectangle(){
side1=getInput("Enter length of rectangle:");
side2=getInput("Enter breadth of rectangle:");
void area(){
double area=side1*side2;
System.out.println("Area of the Rectangle =" +area);}
class triangle extends Shape{
triangle(){
side1=getInput("Enter base of the triangle:");
side2=getInput("Enter height of the triangle:");
void area(){
double area=side1*side2/2;
System.out.println("Area of the triangle=" +area);}
class circle extends Shape{
circle(){
```

```

side1=getInput("Enter the radius of the circle:");
void area(){
double area=Math.PI*side1*side1;
System.out.println("Area of the circle="+area);}

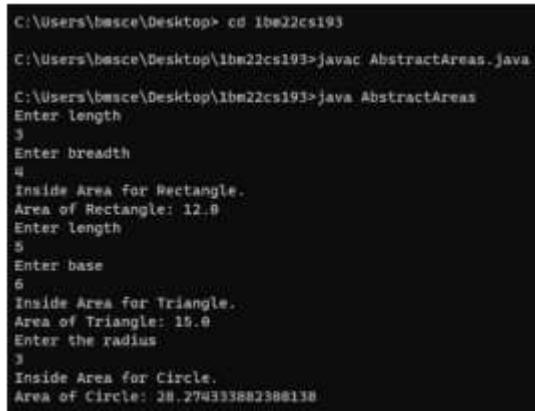
class Main{
public static void main(String args[]){
Rectangle rectangle= new Rectangle();
triangle Triangle=new triangle();
circle Circle=new circle();

rectangle.area();
Triangle.area();
Circle.area();

}
}

```

output:-



```

C:\Users\bmsc1\Desktop> cd lbe22cs193
C:\Users\bmsc1\Desktop\lbe22cs193>javac AbstractAreas.java
C:\Users\bmsc1\Desktop\lbe22cs193>java AbstractAreas
Enter length
3
Enter breadth
4
Inside Area for Rectangle.
Area of Rectangle: 12.0
Enter length
5
Enter base
6
Inside Area for Triangle.
Area of Triangle: 15.0
Enter the radius
3
Inside Area for Circle.
Area of Circle: 28.27433882388138

```

## 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 Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks: a) Accept deposit from customer and update the balance. b) Display the balance. c) Compute and deposit interest d) Permit withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

Code:-

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

```
class Account {
```

```

String customerName;
long accountNumber;
String accountType;
double balance;

public Account(String customerName, long accountNumber, String accountType, double balance) {
    this.customerName = customerName;
    this.accountNumber = accountNumber;
    this.accountType = accountType;
    this.balance = balance;
}

public void deposit(double amount) {
    balance += amount;
    System.out.println("Deposit of $" + amount + " successful. Updated balance: $" + balance);
}

public void displayBalance() {
    System.out.println("Account Balance: $" + balance);
}
}

class CurAccount extends Account {
    double minBalance;
    double serviceCharge;

    public CurAccount(String customerName, long accountNumber, double balance, double minBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minBalance = minBalance;
        this.serviceCharge = serviceCharge;
    }

    public void checkMinBalance() {
        if (balance < minBalance) {
            balance -= serviceCharge;
            System.out.println("Minimum balance not maintained. Service charge of $" + serviceCharge +
" imposed.");
            displayBalance();
        }
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. Withdrawal failed.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);
            checkMinBalance();
        }
    }
}

```

```
    }
}

class SavAccount extends Account {
    double interestRate;

    public SavAccount(String customerName, long accountNumber, double balance, double
interestRate) {
        super(customerName, accountNumber, "Savings", balance);
        this.interestRate = interestRate;
    }

    public void computeInterest() {
        double interest = balance * (interestRate / 100);
        balance += interest;
        System.out.println("Interest computed and deposited: $" + interest);
        displayBalance();
    }

    public void withdraw(double amount) {
        if (amount > balance) {
            System.out.println("Insufficient funds. Withdrawal failed.");
        } else {
            balance -= amount;
            System.out.println("Withdrawal of $" + amount + " successful. Updated balance: $" +
balance);
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        CurAccount currentAccount = new CurAccount("John Doe", 123456789, 1000, 500, 10);
        SavAccount savingsAccount = new SavAccount("Jane Doe", 987654321, 2000, 5);

        int choice;
        do {
            System.out.println("\nSelect an option:");
            System.out.println("1. Deposit");
            System.out.println("2. Display Balance");
            System.out.println("3. Compute Interest (Savings Account only)");
            System.out.println("4. Withdraw");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter amount to deposit: ");

```

```

        double depositAmount = scanner.nextDouble();
        System.out.print("Select account (1. Current, 2. Savings): ");
        int accountType = scanner.nextInt();
        if (accountType == 1) {
            currentAccount.deposit(depositAmount);
        } else if (accountType == 2) {
            savingsAccount.deposit(depositAmount);
        } else {
            System.out.println("Invalid account type.");
        }
        break;
    case 2:
        System.out.print("Select account (1. Current, 2. Savings): ");
        int accType = scanner.nextInt();
        if (accType == 1) {
            currentAccount.displayBalance();
        } else if (accType == 2) {
            savingsAccount.displayBalance();
        } else {
            System.out.println("Invalid account type.");
        }
        break;
    case 3:
        if (savingsAccount instanceof SavAccount) {
            ((SavAccount) savingsAccount).computeInterest();
        } else {
            System.out.println("Invalid option for current account.");
        }
        break;
    case 4:
        System.out.print("Enter amount to withdraw: ");
        double withdrawAmount = scanner.nextDouble();
        System.out.print("Select account (1. Current, 2. Savings): ");
        int accTyp = scanner.nextInt();
        if (accTyp == 1) {
            currentAccount.withdraw(withdrawAmount);
        } else if (accTyp == 2) {
            savingsAccount.withdraw(withdrawAmount);
        } else {
            System.out.println("Invalid account type.");
        }
        break;
    case 5:
        System.out.println("Exiting the program. Thank you!");
        break;
    default:
        System.out.println("Invalid choice. Please enter a valid option.");
    }
} while (choice != 5);

scanner.close();

```

```
}
```

Output:-

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The application displays a series of interactions with a banking system:

- First interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 3
- Interest computed and deposited: INR 2881.05.0
- Account Balance: INR 4282285.0
- Second interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 2
- Select account (1. Current, 2. Savings): 2
- Account Balance: INR 4282285.0
- Third interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 3
- Interest computed and deposited: INR 218110.25
- Account Balance: INR 4412315.25
- Fourth interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 4
- Amount to withdraw: 288000
- Select account (1. Current, 2. Savings): 2
- Withdrawal of INR 288000.0 successful. Updated balance: INR 4212315.25
- Fifth interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 2
- Select account (1. Current, 2. Savings): 2
- Account Balance: INR 4212315.25
- Sixth interaction:
  - Select an option:
  - 1. Deposit
  - 2. Display Balance
  - 3. Compute Interest (Savings Account only)
  - 4. Withdraw
  - 5. Exit
- Enter your choice: 5
- Exiting the program. Thank you!

At the bottom left, the path is shown as C:\Users\oracle\Desktop\11m226s195>.

```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\oracle\Desktop>cd lhm22cs195
C:\Users\oracle\Desktop\lhm22cs195>javac Bank.java
C:\Users\oracle\Desktop\lhm22cs195>java Bank

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 1
Enter amount to deposit: 500000
Select account (1. Current, 2. Savings): 1
Deposit of INR 500000.0 successful. Updated balance: INR 500000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 2
Select account (1. Current, 2. Savings): 1
Account Balance: INR 500000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 3
Interest computed and deposited: INR 100.0
Account Balance: INR 500100.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 4
Enter amount to withdraw: 50000
Select account (1. Current, 2. Savings): 1
Withdrawal of INR 50000.0 successful. Updated balance: INR 450000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 2
Select account (1. Current, 2. Savings): 1
Account Balance: INR 450000.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 1
Enter amount to deposit: 4000000
Select account (1. Current, 2. Savings): 2
Deposit of INR 4000000.0 successful. Updated balance: INR 4002100.0

Select an option:
1. Deposit
2. Display Balance
3. Compute Interest (Savings Account only)
4. Withdraw
5. Exit

Enter your choice: 2
Select account (1. Current, 2. Savings): 2
Account Balance: INR 4002100.0

```

## LAB 6

```

package CIE;

public class Student {

    public String usn;

    public String name;

    public int sem;

    public Student(String usn, String name, int sem) {

        this.usn = usn;

        this.name = name;

        this.sem = sem;

    }

}

package CIE;

```

```
public class Internals extends Student {  
    public int[] internalMarks;  
  
    public Internals(String usn, String name, int sem, int[] internalMarks) {  
        super(usn, name, sem);  
        this.internalMarks = internalMarks;  
    }  
}  
  
package SEE;  
  
22  
  
import CIE.Student;  
  
public class External extends Student {  
    public int[] seeMarks;  
  
    public External(String usn, String name, int sem, int[] seeMarks) {  
        super(usn, name, sem);  
        this.seeMarks = seeMarks;  
    }  
}  
  
import CIE.Internals;  
  
import SEE.External;  
  
import java.util.Scanner;  
  
public class FinalMarks {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of students: ");  
        int n = scanner.nextInt();  
        Internals[] cieStudents = new Internals[n];  
        External[] seeStudents = new External[n];  
        for (int i = 0; i < n; i++) {  
            System.out.println("Enter details for CIE of student " + (i + 1));  
            System.out.print("USN: ");  
            String usn = scanner.next();
```

```
System.out.print("Name: ");

String name = scanner.next();

System.out.print("Semester: ");

int sem = scanner.nextInt();

int[] cieMarks = new int[5];

System.out.print("Enter CIE marks for 5 courses: ");

for (int j = 0; j < 5; j++) {

23

cieMarks[j] = scanner.nextInt();

}

cieStudents[i] = new Internals(usn, name, sem, cieMarks);

}

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

System.out.println("Enter details for SEE of student " + (i + 1));

System.out.print("USN: ");

String usn = scanner.next();

System.out.print("Name: ");

String name = scanner.next();

System.out.print("Semester: ");

int sem = scanner.nextInt();

int[] seeMarks = new int[5];

System.out.print("Enter SEE marks for 5 courses: ");

for (int j = 0; j < 5; j++) {

seeMarks[j] = scanner.nextInt();

}

seeStudents[i] = new External(usn, name, sem, seeMarks);

}

System.out.println("\nFinal Marks of Students:");

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

System.out.println("\nDetails of Student " + (i + 1));

System.out.println("USN: " + cieStudents[i].usn);
```

```
System.out.println("Name: " + cieStudents[i].name);
System.out.println("Semester: " + cieStudents[i].sem);
System.out.println("CIE Marks: ");
for (int j = 0; j < 5; j++) {
    System.out.print(cieStudents[i].internalMarks[j] + " ");
}
System.out.println("\nSEE Marks: ");
```

24

```
for (int j = 0; j < 5; j++) {
    System.out.print(seeStudents[i].seeMarks[j] + " ");
}
}
```

Output:-

```

Marks in :5subject is45
Marks in :1subject is78
Marks in :1subject is79
Marks in :1subject is100
Marks in :1subject is88
Marks in :1subject is91
Enter the usn:
2
Enter the Name:
B
Enter the Sem:
1
Enter marks in 5 subjects:
Marks in :1subject is
45
Marks in :2subject is
44
Marks in :3subject is
35
Marks in :4subject is
58
Marks in :5subject is
32
Enter marks in 5 subjects:
Marks in :1subject is
45
Marks in :2subject is
58
Marks in :3subject is
45
Marks in :4subject is
29
Marks in :5subject is
44
Name is:B
USN is:2
Sem is:1
Internal details
Marks in :1subject is45
Marks in :2subject is44
Marks in :3subject is35
Marks in :4subject is58
Marks in :5subject is32
External details
Marks in :1subject is45
Marks in :2subject is50
Marks in :3subject is45
Marks in :4subject is29
Marks in :5subject is44
Marks in :2subject is90
Marks in :2subject is94
Marks in :2subject is80
Marks in :2subject is79
Marks in :2subject is76

```

```

External details
Marks in :1subject is45
Marks in :2subject is50
Marks in :3subject is45
Marks in :4subject is29
Marks in :5subject is44
Marks in :2subject is90
Marks in :2subject is94
Marks in :2subject is80
Marks in :2subject is79
Marks in :2subject is76

```

## LAB 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age=father’s age

Code:-

```

import java.util.Scanner;
class WrongAge extends Exception{
WrongAge(String error){
System.out.println(error);
}
}

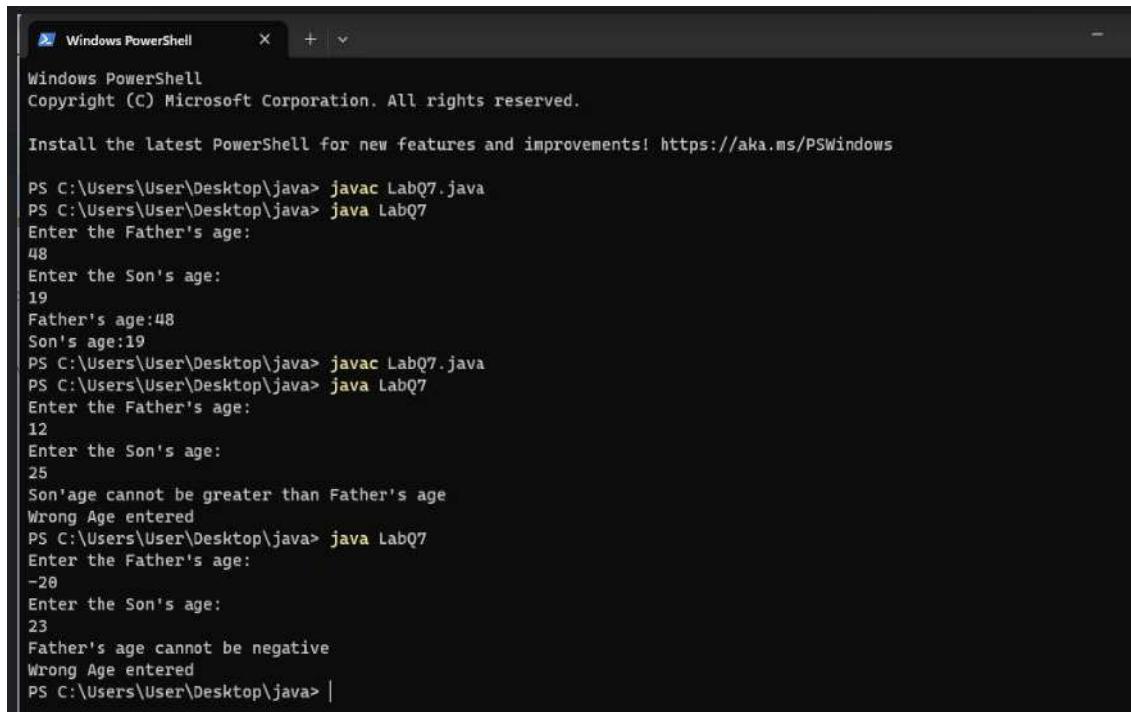
class Father{
int age;
Father(int age) throws WrongAge{
if(age<0)
throw new WrongAge("Father's age cannot be negative");
this.age=age;
}
}

class Son extends Father{
int age;
Son(int age,int s_age) throws WrongAge{
super(age);
if(s_age>=age)
throw new WrongAge("Son's age cannot be greater than Father's age");
this.age=s_age;
}
}

class LabQ7{
public static void main(String args[]){
Scanner sc=new Scanner(System.in);
try{
System.out.println("Enter the Father's age:");
int f_age=sc.nextInt();
System.out.println("Enter the Son's age:");
int s_age=sc.nextInt();
Son a=new Son(f_age,s_age);
System.out.println("Father's age:"+f_age);
System.out.println("Son's age:"+s_age);
}
catch(WrongAge e){
System.out.println("Wrong Age entered");}
catch(Exception ee){
System.out.println("Unexpected error :" +ee);}
}
}

```

output:-



A screenshot of a Windows PowerShell window titled "Windows PowerShell". The window shows the following Java program execution:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User\Desktop\java> javac LabQ7.java
PS C:\Users\User\Desktop\java> java LabQ7
Enter the Father's age:
48
Enter the Son's age:
19
Father's age:48
Son's age:19
PS C:\Users\User\Desktop\java> javac LabQ7.java
PS C:\Users\User\Desktop\java> java LabQ7
Enter the Father's age:
12
Enter the Son's age:
25
Son's age cannot be greater than Father's age
Wrong Age entered
PS C:\Users\User\Desktop\java> java LabQ7
Enter the Father's age:
-20
Enter the Son's age:
23
Father's age cannot be negative
Wrong Age entered
PS C:\Users\User\Desktop\java> |
```

## LAB 8

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

Code:-

```
class DisplayThread extends Thread {
    private String message;
    private int interval;
    private boolean running = true;
    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
    }
    public void run() {
        while (running) {
            System.out.println(message);
            try {
                Thread.sleep(interval);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
    public void stopThread() {
        running = false;
    }
}
```

```
public class ThreadExample {  
    public static void main(String[] args) {  
        DisplayThread bmsThread = new DisplayThread("BMS College of Engineering", 10000);  
        DisplayThread cseThread = new DisplayThread("CSE", 2000);  
        bmsThread.start();  
        cseThread.start();  
        System.out.println("Press Enter to stop the threads...");  
        try {  
            System.in.read();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
        bmsThread.stopThread();  
        cseThread.stopThread();  
    }  
}
```

Output:

```
PS C:\Users\BMSCE\Desktop\1bm22cs195> java Thread2  
Press Enter to stop the threads...  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
CSE  
BMS College of Engineering  
CSE  
CSE  
CSE  
CSE  
CSE  
PS C:\Users\BMSCE\Desktop\1bm22cs195> |
```

## LAB 9

Lab 9 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 ArithmeticException. Display the exception in a message dialog box.

Code:

CODE:

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

class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // text label
        JLabel jlab = new JLabel("Enter the divider and divident:");

        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);

        // calc button
        JButton button = new JButton("Calculate");

        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
```

```
// add in order :)

jfrm.add(err); // to display error bois

jfrm.add(jlab);

jfrm.add(ajtf);

jfrm.add(bjtf);

jfrm.add(button);

jfrm.add(alab);

jfrm.add(blab);

jfrm.add(anslab);

ActionListener l = new ActionListener() {

    public void actionPerformed(ActionEvent evt) {

        System.out.println("Action event from a text field");

    }

};

ajtf.addActionListener(l);

bjtf.addActionListener(l);

button.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent evt) {

        try{

            int a = Integer.parseInt(ajtf.getText());

            int b = Integer.parseInt(bjtf.getText());

            int ans = a/b;

            alab.setText("\nA = " + a);

            blab.setText("\nB = " + b);

            anslab.setText("\nAns = " + ans);

        }

        catch(NumberFormatException e){

            alab.setText(" ");

        }

    }

}
```

```
blab.setText("““);
anslab.setText("““);
err.setText("“Enter Only Integers!“);
}

catch(ArithmeticException e){
    alab.setText("““);

    blab.setText("““);
    anslab.setText("““);
    err.setText("“B should be NON zero!“);
}

}

});

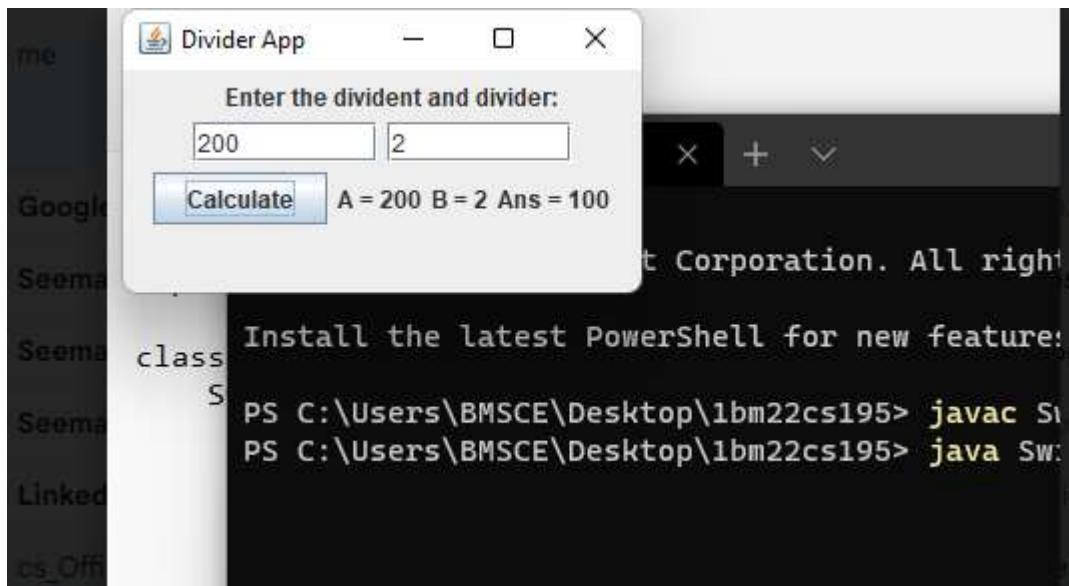
// display frame
jfrm.setVisible(true);
}

public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){

        public void run(){
            new SwingDemo();
        }
    });
}

}

Ouput:-
```



## LAB 10

Demonstrate Inter process Communication and deadlock

Code:

IPC

```
class Q {  
    int n;  
    boolean valueSet = false;  
    synchronized int get() {  
        while(!valueSet)  
            try {  
                wait();  
            } catch(InterruptedException e) {  
                System.out.println("InterruptedException caught");  
            }  
        System.out.println("Got: " + n);  
        33  
        valueSet = false;  
        notify();  
        return n;  
    }  
    synchronized void put(int n) {
```

```
while(valueSet)
try {
wait();
} catch(InterruptedException e) {
System.out.println("InterruptedException caught");
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
notify();
}
}
```

```
class Producer implements Runnable {
Q q;
Producer(Q q) {
this.q = q;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(i<15) {
q.put(i++);
}
}
}
```

34

```
class Consumer implements Runnable {
Q q;
Consumer(Q q) {
this.q = q;
new Thread(this, "Consumer").start();
```

```

}

public void run() {
    int i=0;
    while(i<15) {
        int r=q.get();
        i++;
    }
}

class PCFixed {
    public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
    }
}

```

Output:



Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```

PS C:\Users\BMSCE\Desktop\1bm22cs195> Java Deadlock.java
PS C:\Users\BMSCE\Desktop\1bm22cs195> Java Deadlock
MainThread entered A.foo
RacingThread entered B.bar
MainThread trying to call B.last()
Inside A.last
Back in main thread
RacingThread trying to call A.last()
Inside A.last
Back in other thread
PS C:\Users\BMSCE\Desktop\1bm22cs195> ^Z

```