# BMS COLLEGE OF ENGINEERING

**(AUTONOMOUS COLLEGE, AFFILIATED TO VTU)**

**Bull Temple Road, Basavangudi, Bangalore -560019**

**2023-2024**

# Lab Observation

IN

## Object Oriented Programming

## In Java

BY


**SETU MISHRA**

USN-1BM22CS250

# INDEX

Name Setu Mishra Std Java Lab(OOPs) Sec

1BM22CS250

Roll No. Subject OOPs in Java School/College BMSCE

School/College Tel. No. Parents Tel. No.

| Sl. No. | Date | Title | Page No. | Teacher Sign / Remarks |
|---|---|---|---|---|
| 1. | 5/12/23 | Sample program | | |
| 2. | 12/12/23 | program 1 - Quadratic eqn | 10 | |
| 3. | 19/12/23 | program 2 - Student SGPA | 10 | 19/12/2023 |
| 4. | 25/12/23 | program 3 - Author book | 10 | 26/12/23 |
| 5. | 02/01/24 | program 4 - Shape superclass | 10 | 2/1/2024 |
| 6. | 09/01/24 | program 5 - Bank | 10 | |
| 7. | 16/01/24 | Strings - Extra programs | 10 | 23/1/2024 |
| 8. | 23/01/24 | Packages - Lab 6. | 10 | 30.1.24 |
| 9. | 30/01/24 | Lab 7 - FatherSon | 10 | 1/2/24 |
| 10. | 6/02/24 | Lab 8 - 2 threads | 10 | 6-2-24 |
| 11. | 13/02/24 | Lab 10 → IPC → Deadlock | 10 | 13-2-24 |
| 12. | 20/02/24 | Lab-9 - Division Main | | 20/2/2024 |

5/12/23

Date __/__/__
Page _____

# Program to say hello world..

```
import java.util.*;
public class Demo1 {
    public static void main (String args[])
    {
        System.out.println ("hello world");
    }
}
```

output: hello world

# Program to add 2 numbers

```
import java.util.*;
class Add{
    public static void main (String args[])
    {
        int n1=5, n2=10, sum;
        sum= n1+n2;
        System.out.println (sum);
    }
}
```

output: 15

# Program to multiply 2 numbers

```
import java.util.*;
class multiply{
    public static void main (String args[])
    {
        int n1=5, n2=10, mul;
        mul = n1*n2;
        System.out.println (mul);
    }
}
```

output: 50

Date
Page

# Program to compare 2 numbers.

```
import java.util.*;
class comp {
    public static void main (String args[])
    {
        int a=10;
        int b=25;
        if (a>b)
        {
            System.out.println ("A is greater");
        }
        else if (a<b)
        {
            System.out.println("B is greater");
        }
        else
        {
            System.out.println ("Both are
                                 equal");
        }
    }
}
```

Output: b is greater

12/12/23

# Lab-1

Date __/__/__
Page. _____

(1) Develop a java program that prints all real solution of the quadratic eq. $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula. If the discriminant $b^2-4ac$ is -ve, display a message stating that there are no real solutions.

```
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");
        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 ("Root are
                real and equl");
    System.out. println("Root 1 =
                Root 2=" +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(" Root 1 = "+r1+"
                Root 2 = "+r2);
}
else if (d<0)
{
    System.out.println("Roots are
                imaginary");
    r1=(-b) /(2*a);
    r2= Math.sqrt (-d)/(2*a);
    System.out.println("Root 1 = "+r1+
                " +i " +r2);
    System.out.println(" Root 1 ="+r1+
                " -i" +r2);
}
}
class Quadratic Main
{
    public static void main (String args[])
{
```

Date __/__/__
Page. _____

```
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

Rectangle.java

```
class rectangle{
    public static void main(String args[]){
        int a,b;
        a= Integer.parseInt(args[0]);
        b= Integer.parseInt(args[1]
```

output:

Setu Mishra
1BM22CS250

1. Enter the coefficient a, b, c.
   3 1 2
   roots are imaginary
   root 1 = 0.0 + i0.7993052538854537
   root 2 = 0.0 - i0.7993052538854531

2. Enter the coefficient a, b, c
   1 4 1
   root 1 = -2.0 + iNaN
   root 2 = -2.0 - iNaN

3. Enter the coefficients a, b, c.
   1 2 1
   roots are real and equal
   root 1 = root 2 = -1.0

Date / /
Page

(2) To write a program in Java to find the area of a rectangle and verify the same with various inputs (length, breadth)

```
import java.util.*;
class Rectangletrea{
public static void main (String args[])
{
    int length, breadth;
    length= Integer.parseInt(args[0]);
    breadth= Integer.parseInt(args[1]);
    int area= length * breadth;
    System.out.println("length of
    rectangle =" + length);
    System.out.println("breadth of
    rectangle =" + breadth);

    System.out.println("area of
    rectangle =" + area);
}
}
```

O/p: length of rectangle = 10
breadth of rectangle = 12
area of rectangle = 120

Setu Mishra
1BM22CS250

19/12/23

# Lab-2

Date __/__/__
Page _____

Q) Develop a java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

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

    Student(){
        int i;
        subjects = new Subject [8];
        for( i=0; i<=8; i++) {
            subject[i] = new Subject();
            s= new Scanner(System.in);
    }
    void getStudentDetails(){
        System.out.println("Enter student
                            name :");
    this.name = s.next();
        System.out.println("Enter USN:");
        this.usn = s.next();
    }
```

```
void getMarks(){
    for(int i=0; i<=8; i++){
    System.out.println("Enter details
            for subject "+(i+1));
    System.out.println("Enter marks");
    subjects[i].subjectMarks = s.nextInt();
    System.out.println("Enter credits:");
    subjects[i].credits = s.nextInt();

    if(subjects[i].subjectMarks
                                >=90)
    {
      subjects[i].grade=10;
    }
    elseif(subjects[i].subjectMarks>=80)
    {
      subjects[i].grade=9;
    }
    elseif(subjects[i].subjectMarks >=70)
    {
      subjects[i].grade=8;
    }
    elseif(subjects[i].subjectMarks>=60)
    {
      subjects[i].grade=7;
    }
    elseif(subject[i].subjectMarks>=50){
      subjects[i].grade=6;
    }
    elseif(subject[i].subjectMarks>=40){
      subjects[i].grade=5;}
    else{
          subject[i].grade=0;}
}
}
```

Date __/__/__
Page _____

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

}
void displayResult(){
    System.out.println("Student details");

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

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

    System.out.println("SGPA:" +SGPA);

}
public class Main{
    public static void main(String[]args){
        Student s1 = new Student();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();

    }
}
```

Setu Mishra
1BM22CS250

Date / /
Page

output:

Enter student name: Aayra
Enter student usn: 1BM22CS250
Enter details for subject 1

Enter marks: 98
Enter credits: 4
Enter marks: 89
Enter credits: 3
Enter marks: 91
Enter credits: 3
Enter marks: 87
Enter credits: 3
Enter marks: 78
Enter credits: 2
Enter marks: 87
Enter credits: 3
Enter marks: 93
Enter credits: 3
Enter marks: 89
Enter credits: 3
Enter marks: 76
Enter credits: 3
Student details
Name: Aayra
USN: 1BM22CS250
SGPA: 9.185185

19/12/2023

26/12/23

Date __/__/__
Page _____

## Lab program 3

(8) 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.*;
class Books{
        String name;
        String author;
        int price;
        int numPages;

    Books(String name, String author,
          int price, int numPages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = numPages;
    }
    public String toString()
    {
        String name, author, price, numPages;
        name = "Book name:" + this.name +
                "\n";
```

```
author="Author name :" +this.author+
                        "\n";
price= "price:" + this.price + "\n";
numPages= "Number of pages:" +
          this.numPages +"\n";

return name+author+price+numPages;
}
}

public class Main{
    public static void main (String[] args)
    {
        Scanner sc= new Scanner(System.in);
        int n;
        int i;
        String name;
        String author;
        int price;
        int numPages;
        System.out.println(" Enter the
                number of books :\n");
        n=sc.nextInt();
        Books b[];
        b= new Books [n];
        for(i=0;i<n;i++){
            System.out.println("Enter the
                name of the book");
        name= sc.next();
        System.out.println("Enter the
                name of the author");
        author=sc.next();
```

Date / /
Page

```
System.out.println("Enter the price
Of the book");
price = sc.nextInt();
System.out.println("Enter the
number of pages of the book");
numPages = sc.nextInt();

b[i] = new Books(name, author,
                 price, numPages);
}
System.out.println("Displaying the
book details:");
for(i=0; i<n; i++){

System.out.println("Book details
are:)" + (i+1) + "\n" +
b[i].toString());

}
}
}
```

Setu Mishra
1BM22CS250

Output

```
Enter the number of books:
3
Enter the name of the book
Apple
Enter the name of the author
Alice
Enter the price of the book
250
Enter the number of pages of the book
200
```

Enter the name of the book
Banana
Enter the name of the author
Enid
Enter the price of the book
700
Enter the number of pages of the book
1000
Enter the name of the book
Famous
~~Enter the author of the bo~~
Enter the name of the author
Thea
Enter the price of the book
450
Enter the number of pages of the book
150

Displaying the book details:

Book details are: 1
Book name: Apple
Author: Alice
Price: 250
Number of pages: 200

Book details are: 2
Book name: Banana
Author: Enid
Price: 700
Number of pages: 1000

Date __/__/__
Page _____

Book details are:-3
Book name:- Famous
Author:- Thea
Price:- 450
Number of pages:- 150

26/12/23

02/01/24

Date __/__/__
Page

# Lab program-4

Exec

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

```
import java.util.Scanner;

class InputScanner {
    public static Scanner scanner = new
                    Scanner(System.in);

    public static int getIntInput(String
                    message){
    System.out.println(message);
    return scanner.nextInt();
    }
}

abstract class Shape extends InputScanner{
    int side1;
    int side2;

    public static void printArea();
}
```

Date __/__/__
Page _____

```
class Rectangle extends Shape {
    printArea(){
        side1 = getIntInput("Enter length of
            the rectangle:");
        side2 = getIntInput("Enter width of
            the rectangle:");
        int area = side1 * side2;
        System.out.println("Area of the
            rectangle:" + area);
    }
}

class Triangle extends Shape{
    printArea(){
        side1 = getIntInput("Enter the
            base of the triangle:");
        side2 = getIntInput("Enter the
            height of the triangle:");
        double area = 0.5 * side1 * side2;
        System.out.println("Area of the
            triangle:" + area);
    }
}

class Circle extends Shape{
    printArea(){
        side1 = getIntInput("Enter radius
            of the circle:");
```

```
        double area = Math.PI * side1 * side2;
        System.out.println("Area of the
                circle:" + area);
    }
}

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

        Rectangle rectangle = new Rectangle();
        Triangle triangle = new Triangle();
        Circle circle = new Circle();

        rectangle.printArea();
        System.out.println();

        triangle.printArea();

        System.out.println();
        circle.printArea();

    }
}
```

Output:

Sela Mishra
IBM22CS250

```
Enter length of the rectangle: 22
Enter width of the rectangle: 22
Area of the rectangle: 484

Enter the base of the triangle: 22
Enter the height of the triangle: 222
```

Date / /
Page

Area of the triangle : 2442.0

Enter the radius of the circle 22

Area of the circle : 1520.530844933745 97

21/2024

---

09/01/24

Lab program 5

Q) 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-acc and Sav-acc 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

Date __/__/__
Page _____

(d) Permit withdrawal and update the balance
Check for the minimum balance, Impose penalty if necessary and update the balance.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

class Account {
    String customerName;
    long accountNumber;
    String accountType;
    double balance;

    public Account(String customerName,
        long accountNumber, String
        accountType){
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = 0.0;
    }

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

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

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

class CurrAcct extends Account{
    double minBalance = 500;

    public CurrAcct(String customerName, long accountNumber){
        super(customerName, accountNumber, "Current");
    }
    public void checkMinBalance(){
        if (balance < minBalance){
            double penalty = 50.0;
            balance -= penalty;
            System.out.println("Minimum balance not maintained. Penalty imposed. Updated balance: $" + balance);
        }
    }
}
```

Date / /
Page

```
class SavAcct extends Account {
    double interestRate = 0.05;

    public SavAcct (String customerName,
                    long accountNumber) {
        super(customerName, accountNumber,
              "Savings");
    }

    public void computeAndDepositInterest
                    (int timePeriod) {
        double interest = balance *
                interestRate * timePeriod;
        balance += interest;
        System.out.println("Interest
        computed and deposited. Updated
        balance is $" + balance);
    }

    public void withdraw (double amount)
    {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawal
            successful. Update balance is $"
                    + balance);
        }
    }
}

public class Bank {
    public static void main (String[]
    args) throws IOException {
```

```
BufferedReader reader = new
                BufferedReader(
                new InputStream
                Reader(System.
                in));
System.out.print("Enter S for saving
and C for current account : ");

String accountType = reader.readLine();

if (accountType.equalsIgnoreCase("C"))
{
    System.out.print("Enter your name");

long accNo = Long.parseLong(reader.
                readLine());

CurrAcct currentAccount = new
                CurrAcct(name,
                accNo);
System.out.println("Do you want to
deposit? Type 'yes':");

String depositChoice = reader.readLine();

if (depositChoice.equalsIgnoreCase
("yes")){
    System.out.println("Enter the amount
        to deposit.");
double depositAmount = Double.
                parseDouble(reader.
                readLine());
```

Date __/__/__
Page _____

```
currentAccount.deposit(depositAmount);
}

System.out.println("Do you want to withdraw? Type 'yes':");

String withdrawChoice = reader.readLine();
if (withdrawChoice.equalsIgnoreCase("yes")) {
    System.out.println("Enter the amount to withdraw:");
    double withdrawAmount = Double.parseDouble(reader.readLine());

    currentAccount.withdraw(withdrawAmount);
}
    currentAccount.displayBalance();
    currentAccount.checkMinBalance();
} else if
(accountType.equalsIgnoreCase("S")) {
    System.out.println("Enter your name");

    String name = reader.readLine();
    System.out.println("Enter your account number:");
    long accNo = Long.parseLong(reader.readLine());

    SavAcct savingsAccount = new SavAcct(name, accNo);
```

Date __/__/__
Page

```
System.out.println("Do you want to deposit? Type 'yes':");
String depositChoice = reader.readLine();

if (depositChoice.equalsIgnoreCase("yes")) {
System.out.println("Enter the amount to deposit:");
double depositAmount = Double.parseDouble(reader.readLine());
SavingsAccount.deposit(depositAmount);
}

System.out.println("Enter the time period (in years) the money is kept:");
int timePeriod = Integer.parseInt(reader.readLine());

savingsAccount.computeAndDepositInterest(timePeriod);

System.out.println("Do you want to withdraw? Type 'yes':");
```

```
String withdrawChoice = reader.readLine();
if (withdrawChoice.equalsIgnoreCase("yes"))
{
    System.out.println("Enter the amount to withdraw: ");
    double withdrawAmount = Double.parseDouble(reader.readLine());
    savingsAccount.withdraw(withdrawAmount);
}
savingsAccount.displayBalance();
} else {
    System.out.println("Invalid account type. Please enter 'c' for current or 's' for savings account.");
}
reader.close();
}
}
```

Seetu Mishra
IBM22CS250

Output: (Savings)

→ Enter s for saving and c for current account: s

Enter your name: Seetu

Enter your account number: 123

Date __/__/__
Page _____

Do you want to deposit? Type 'yes': Yes

Enter the amount to deposit: 600

Deposit successful. Updated balance: $600.0

Enter the time period (in years) the money is kept: 3
Interest computed and deposited.
Updated balance: $ 690.0

Do you want to withdraw? Type 'yes': Yes

Enter the amount to withdraw: 600
Withdraw successful. Updated balance: $90.0
Current balance: $90.0

→ (Current)

Enter S for saving and C for current account: C
Enter your name: Selu

Enter your account number: 1234

Do you want to deposit? Type 'yes': Yes

Enter the amount to deposit: $4000.0

Deposit successful. Updated balance: $4000.0

Date __/__/__
Page _____

Do you want to withdraw? Type 'yes': yes

Enter the amount to withdraw: 3750

Withdrawal successful. Updated balance: $250.0

Current balance: $250.0

Minimum balance not maintained.
Penalty imposed. Updated balance: $200.0

—— x ——

Satu Mishra
1BM22CS250

23/11/2024.

16/04/24

# Strings (Extra programs)

Date __/__/__
Page ____

Seta Mishra
1BM22CS290

## Outputs:

(1) Hello World
Java
Hello
String Buffer Demo

(2) Length of the string: 11
String literals are equal: true
Concatenated string: Hello World

(3) Converted to String: 42

(4) Extracted: Bmsce

(5) Bytes: 72 101 108 111 32 87 111 114 108
Chars: Hello World

(6) Bmsce equals Bmsce → true
Bmsce equals College → false
Bmsce equals BMSCE → false
Bmsce equals IgnoreCase BMSCE → true

(7) Substring is matched

(8) true
false

(9) true
false

(10) using equals(): true
using ==: false

Date / /
Page

(11) Sorted Words: [apple, ball, cat, dog, eat, free, gun, hen, ice, jug, kite, life, man, net, orange, parrot, queen, ring, star, tree, umbrella, van, watch, xmas, yatch, zu]

(12) Sorted Numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

(13) Original String: Thwas was a test Thwas was too.

String after replacement:
This is a test. This is, too.

(14) Concatenated String: helloworld

(15) Original: Welcome to Bmsce College
Replacement: Welcome to Bmsce Commege.

(16) Original: "Hello Friends"
Trimmed: "Hello Friends"

(17) Student Records:
Reg. Number: 1
Full Name: Selu
Semester: 3
CGPA: 9.4

Reg. Number: 2
Full Name: Shantanu
Semester: 3

Date __/__/__
Page _____

CGPA: 8.6

Reg. Number: 3
Semester: 3 ← Full Name: Aayra
CGPA: 9.8

Reg. Number: 4
Full Name: Santosh
Semester: 4
CGPA: 6.5

Student Records after sorting by CGPA

Reg. Number: 3
CGPA : 9.8
Reg. Number: 1
CGPA : 9.4
Reg. Number: 2
CGPA : 8.6
Reg. Number: 4
CGPA : 6.5

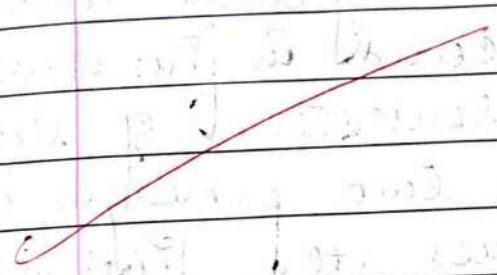
(18) setLength (3): Hel
charAt (2): l
setCharAt (1, 'X'): HXl
getChars (0, 3, dest, 0): HXl
append ("World"): HXl Wo Java red
reverse (): dlr ava Jow lXH
delete (5, 10): dlr a lXH
deleteCharAt (2): dl a lXH
replace (1, 4, "i"): di lXH

Date __/__/__
Page _____

(19) Eagle flies high in the sky
Eagle makes screeching sounds
Hawk flies swiftly through air
Hawk makes a distinctive screech

(20) Circle Area: 78.53981
Circle Perimeter: 31.41592
Triangle Area: 6.0
Triangle Perimeter: 12.0

— x —



Done
23-1-2024

# Lab - 6

Date __/__/__
Page _____

8) Create a package CIE which has two classes - Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student 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.

// Student.java

```
package CIE;
import java.util.*;
public class Student
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStudentDetails()
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Student USN");
```

```
        usn = sc.next();
        System.out.println("Enter
                    Student name");
        name = sc.next;

        System.out.println("Enter semester");
        sem = sc.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("Student usn: "+
                    usn);

        System.out.println("Student name: "+
                    name);
        System.out.println("Student Semester:
                    + sem);
    }
}

// Internals.java
package CE;
import java.util.*;
public class Internals extends Student
{
    protected int marks[] = new int[5];

    public void inputCEmarks()
    {
        Scanner sc = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Enter 5
                    subject marks");
```

```
        marks[i] = sc.nextInt();
    }
  }
}
```

```
// Externals.java
package SEE;
import CIE.Internals;
import java.util.Scanner;

public class Externals extends Internals
{
    protected int marks[];
    protected int finalMarks[];

    public Externals()
    {
        marks = new int[5];
        finalMarks = new int[5];
    }

    public void inputSEEmarks()
    {
        Scanner sc = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Subject" +
                (i+1) + "marks:");
            marks[i] = sc.nextInt();
        }
    }
}
```

```
public void calculateFinalMarks()
{
    for(int i=0; i<5; i++)
    {
        finalMarks[i] = marks[i] + super.marks[i];
    }
}

public void displayFinalMarks()
{
    displayStudentDetails();
    for(int i=0; i<5; i++)
    {
        System.out.println("Subject" +
            (i+1) + ": " + finalMarks[i]);
    }
}

//Main.java
import SEE.Externals
class Main
{
    public static void main(String args[])
    {
        int numStudents = 2;
        External finalMarks[] = new
                External[numStudents];
        for(int i=0; i<numStudents; i++)
        {
            finalMarks[i] = new External();
```

```
finalMarks[i].inputStudentDetails();
System.out.println("Enter CIE marks");
finalMarks[i].inputCIEmarks();
System.out.println("Enter SEE marks");
finalMarks[i].inputSEEmarks();
}
System.out.println("Displaying data:\n");
for(int i=0; i<numStudents; i++)
{
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
}
}
}
```

Output:

Setu Mishra
1BM22CS250

```
Enter student usn
1BM22CS250
Enter student name
Setu
Enter student semester
3

Enter CIE marks
Enter 5 marks: 39
"  : 38
"  : 26
"  : 40
"  : 36
```

Enter SEE marks

Subject 1 marks: 78
" 2 " : 89
" 3 " : 90
" 4 " : 87
" 5 " : 92

Enter student usn
1BM22CS247
enter student name
Aayra
Enter student semester
3

Seta Mishra
1BM22CS250

Enter CIE marks
Enter 5 marks: 39
" : 38
" : 30
" : 30
" : 25

Enter SEE marks
Subject 1 marks: 56
" 2 " : 78
" 3 " : 89
" 4 " : 90
" 5 " : 92

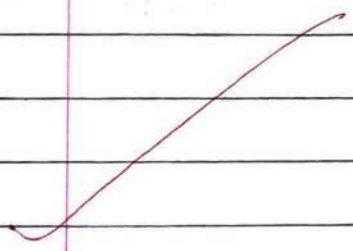
Displaying data
Student name: Seta
Student USN: 1BM22CS250
Student semester: 3

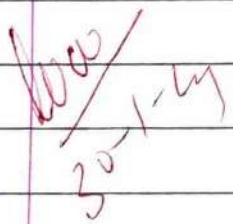
Date __/__/__
Page

Subject 1: 78
Subject 2: 82
Subject 3: 71
Subject 4: 83
Subject 5: 82

Student name: Aayra
Student usn: 1BM22CS247
Student semester: 3

Subject 1: 58
Subject 2: 77
Subject 3: 80
Subject 4: 70
Subject 5: 80





Lab-7

Date __/__/__
Page

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 >= father's age.

```
import java.util.Scanner;
class WrongAge extends Exception {
    public WrongAge (String message) {
        super (message);
    }
}
class InputScanner {
    protected Scanner scanner;

    public InputScanner(){
        scanner=new Scanner(System.in);
    }
    public int nextInt(){
        return scanner.nextInt();
    }
}

class Father extends InputScanner {
    protected int fatherAge;

    public Father() throws WrongAge {
```

Date / /
Page

```
System.out.println("Enter father's age:");
fatherAge = scanner.nextInt();
if (fatherAge < 0)
    throw new WrongAge("Age cannot be negative");
}

public void display(){
    System.out.println("Father's age:" + fatherAge);
}
}

class Son extends Father{
    private int sonAge;

    public Son() throws WrongAge{
    super();
    System.out.println("Enter son's age:");
    sonAge = scanner.nextInt();
    if (sonAge > fatherAge)
        throw new WrongAge("Son's age cannot be greater than father's age");
    elseif (sonAge < 0)
        throw new WrongAge("Age cannot be negative");
    elseif (sonAge == fatherAge)
        throw new WrongAge("Same age not possible");
}
```

```
public void display(){
    super.display();
    System.out.println("Son's age:" +
                        sonAge);
}
}

public class fatherson{
    public static void main(String[] args){
    try{
        Son son = new Son();
        son.display();
    } catch (WrongAge e){
        System.out.println("Error:" +
            e.getMessage());
    }
}
}
```

Seetu Mishra
1BM22CS250

Output

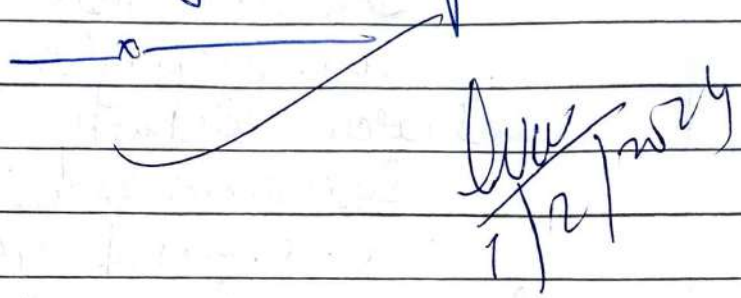
Enter father's age:
48
Enter son's age:
21
Father's age: 48
Son's age: 21.

Enter father's age:
12
Enter son's age:
56
Error: Son's age cannot be greater than father's age.

Date __/__/__
Page _____

Enter father's age:
-12
Error: Age cannot be negative

Enter father's age:
10
Enter son's age:
10
Error: Same age not possible.



Lab-8

Date __/__/__
Page _____

(8) WAP which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
class BMSCE extends Thread
{
  public void run()
  {
    for(int i=1; i<=5; i++)
    System.out.println("BMS College of
    Engineering "+i);
    try{
      Thread.sleep(10000);
    }
    catch(InterruptedException e){
    System.out.println("Interrupted");
  }
}
class CS extends Thread
{
  public void run()
  {
    for int i=1; i<=5; i++)
    System.out.println("CSE"+i);
    try{
      Thread.sleep(2000);
    }
    catch(InterruptedException e){
    System.out.println("Interrupted");
  }
}
```

```
public class BMSCE_CS_ThreadProgram{
    public static void main(String args[])
    {
        BMSCE c1= new BMSCE();
        c1.start();

        CS c1= new CS();
        c1.start();
    }
}
```

Op:

```
BMS College of Engineering 1
BMS College of Engineering 2
CSE 1
CSE 2
CSE 3
CSE 4
CSE 5
BMS College of Engineering 3
BMS College of Engineering 4
BMS College of Engineering 5
```

6.2.23

13/2/24

Date __/__/__
Page _____

## Lab- program 10 (IPC)

```
class Q{
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while (valueSet)
        try{
            System.out.println("\n consumer waiting\n");
            wait();
        }
        catch(InterruptedException e){
            System.out.println("Interrupted exception caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("\nInformate Producer\n");
        notify();
        return n;
    }

    synchronized void put(int n){
        while(valueSet)
        try{
            System.out.println("\n Producer waiting\n");
            wait();
        } catch(InterruptedException e){
            System.out.println("Interrupted exception caught");
        }
```

```
this.n = n;
valueSet = true;
System.out.println("Put : " + n);
System.out.println("\n Intimate Consumer\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++);
  }
}
}
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();
        System.out.println("consumed:" + r);
        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:

Put: 0
Interrupted Consumer
Producer waiting
Press Control-C to stop.
Got: 0
Interrupted Producer
Consumed: 0

Put: 1
Intimate Consumer
~~Consumer~~
Producer waiting
Got: 1
Intimate Producer
Consumed: 1
Put: 2
Intimate Consumer
Producer waiting
Got: 2
Intimate Producer
Consumed: 2
Put: 3
Intimate Consumer
Producer waiting
Got: 3
Intimate Producer
Consumed: 3
Put: 4
Intimate Consumer
Producer waiting
Got: 4
Intimate Producer
Consumed: 4
Put: 5
Intimate Consumer
Producer waiting
Got: 5
Intimate Producer
Consumed: 5
Put: 6
Intimate Consumer

Producer waiting
Got: 6
Initiate Producer
Consumed: 6

.
.
.
.
.

Consumed: 14

— x —

13-2-24

Date __/__/__
Page

# Lab-10

## Deadlock

```
class A{
    Synchronized void foo (B b) {
    String name = Thread.current Thread().
                                getName();

    System.out.println(name + " entered
                                A.foo");
    try{
        Thread.sleep(1000);
    } catch (Exception e){
        System.out.println ("A Interrupted");
    }
    System.out.println (name + "trying to
                                call B.last()");
    b.last();
}
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("B interrupted");
}
```

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

    void last(){
        System.out.println("Inside A.last");
    }
}

class Deadlock implements Runnable
{
    A a= new A();
    B b= new B();
    Deadlock();
    Thread.currentThread().setName("Main Thread");
    Thread t= new Thread(this, "Racing Thread");
    t.start();
    a.foo(b);
    System.out.println("Back in main thread");
}

public void run(){
    b.bar(a);
    System.out.println("Back in other thread");
}
public static void main(String.args[])
{
    new Deadlock();
}
}
```

Date __/__/__
Page _____

Output:

Main Thread entered A.foo
Racing Thread entered B.bar
Racingthread trying to call A.last()
Inside A.last
Back in other thread
MainThread trying to call B.last().
Inside A.last
Back in main thread

13-2-24

# Lab-7

Date __/__/__
Page _____

(8) Wap that creates a user interface to perform 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 and Num2 were not an Integer, the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmaticException. Display the exception in a message dialog box.

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

public class DivisionMain1 extends Frame
        implements ActionListener
{
    TextField num1, num2;
    Button dResult;
    Label outResult;
    String out = " ";
    double resultNum;
    int flag = 0;

    public DivisionMain1()
    {
        setLayout(new FlowLayout());
        dResult = new Button("RESULT");
        Label number1 = new Label("Number1:",
                Label.RIGHT);
```

Date __/__/__
Page _____

```
Label number2 = new Label("Number
        2:", Label.RIGHT);

num1 = new TextField(5);
num2 = new TextField(5);
outResult = new Label("Result :",
                Label.RIGHT);

add(number1);
add(num1);
add(number2);
add(num2);
add(dResult);
add(outResult);

num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent we)
    {
        System.exit(0);
    }
});
}
public void actionPerformed(ActionEvent
                ae)
{
    int n1, n2;
    try
    {
        if(ae.getSource() == dResult)
        {
```

Date __/__/__
Page _____

```
n1 = Integer.parseInt(num1.getText());
n2 = Integer.parseInt(num2.getText());

/* if (ae.getSource() == dResult)

/* if (n2 == 0)
    throw new ArithmeticException(); */

out = n1 + " " + n2;
resultNum = n1 / n2;
out += String.valueOf(resultNum);
repaint();
}
}
catch(NumberFormatException e1)
{
    flag = 1;
    out = "Number Format Exception" + e1;
    repaint();
}
catch(ArithmeticException e2)
{
    flag = 1;
    out = "Divide by 0 exception!" + e2;
    repaint();
}
}
```

```
public void paint (Graphics g)
{
    if (flag == 0)
    g.drawString (out, outResult.
                        getX() +
    outResult.getWidth() >
    outResult.getY() +
    outResult.getHeight() - 8);
    else
    g.drawString (out, 100, 200);
    flag = 0;
}

public static void main (String [] args)
{
    DivisionMain1 dm = new DivisionMain1
                                ();
    dm.setSize (new Dimension (800, 400));
    dm.setTitle ("Division of Integers");
    dm.setVisible (true);
}
}
```

Output:

Number1: [95] Number2: [45]

RESULT Result: 95 45 2.0

20/5/2024

—x—

# PROGRAM-01

**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.**

```
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=Root2= "+r1);
}
else if(d>0)
{
r1=((-b)+(Math.sqrt(d)))/(double)(2*a);
r1=((-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();
}
}
```

## PROGRAM-02

**Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.**

```
import java.util.*;
class Subject
{
int subjectMarks;
int credits;
int grade;
}
class Student
{
Subject subject[];
String name;
String usn;
double sgpa;
```

```
Scanner s;
Student()
{
int i;
subject =new Subject[9];
for(i=0;i<9;i++)
subject[i]=new Subject();
s=new Scanner(System.in);
}
void getStudentDetails()
{
System.out.println("Enter student name");
name=s.next();
System.out.println("Enter student USN");
usn=s.next();
}
void getMarks()
{
for(int i=0;i<9;i++)
{
System.out.println("Enter marks");
subject[i].subjectMarks=s.nextInt();
System.out.println("Enter number of +credits");
subject[i].credits=s.nextInt();
subject[i].grade=(subject[i].subjectMarks/10)+1;
if(subject[i].grade==11)
```

```
subject[i].grade=10;
if(subject[i].grade<=4)
subject[i].grade=0;
}
}
void computeSGPA()
{
int effectiveScores=0;
int totalCredits=0;
for(int i=0;i<9;i++)
{
effectiveScores+=subject[i].grade*subject[i].credits;
totalCredits+=subject[i].credits;
}
sgpa=(double)effectiveScores/(double)totalCredits;
}
}
class Main
{
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);
System.out.println("Student SGPA:"+s1.sgpa);
}
}
```

## PROGRAM-03

**Create a class Book which contains four members: name,author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.**

```
import java.util.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() {
```

```
return "Book name: " + name + "\n" +
"Author name: " + author + "\n" +
"Price: " + price + "\n" +
"Number of pages: " + numPages + "\n";
}
}
class BooksMain {
public static void main(String args[]) {
Scanner s = new Scanner(System.in);
int n;
System.out.println("Enter the number of books: ");
n = s.nextInt();
Book[] books = new Book[n];
for (int i = 0; i < n; i++) {
System.out.println("Enter the name of the book");
String name = s.next();
System.out.println("Enter the author of the book");
String author = s.next();
System.out.println("Enter the price of the book");
int price = s.nextInt();
System.out.println("Enter the pages of the book");
int numPages = s.nextInt();
books[i] = new Book(name, author, price, numPages);
}
for (int i = 0; i < n; i++) {
System.out.println(books[i].toString());
```

```
}
}
}
```

## PROGRAM-04

**Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ).Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method**

printArea( ) that prints the area of the given shape.

```
import java.util.*;
abstract class Shape{
double dim1;
double dim2;
double rad;
Shape(double a,double b)
{
dim1=a;
dim2=b;
}
Shape(double a)
{
rad=a;
}
```

```
abstract void area();
}
class Rectangle extends Shape
{
Rectangle(double a,double b)
{
super(a,b);
}
void area()
{
System.out.println("Area of rectangle is:"+(dim1*dim2));
}
}
class Triangle extends Shape
{
Triangle(double a,double b)
{
super(a,b);
}
void area()
{
System.out.println("Area of triangle is"+(dim1*dim2)/2);
}
}
class Circle extends Shape
{
```

```
Circle(double a)
{
super(a);
}
void area()
{
System.out.println("Area of circle is:"+(3.14*rad*rad));
}
}
class AbstractMain{
public static void main(String args[])
{
Scanner s=new Scanner(System.in);
System.out.println("Enter the dimensions of rectangle");
double l=s.nextInt();
double b=s.nextInt();
System.out.println("Enter the base and height of traingle");
double h1=s.nextInt();
double h2=s.nextDouble();
System.out.println("Enter the radius of circle");
double r=s.nextInt();
Shape sh;
Rectangle rect=new Rectangle(l,b);
Triangle tri=new Triangle(h1,h2);
Circle cir=new Circle(r);
sh=rect;
```

```
sh.area();
sh=tri;
sh.area();
sh=cir;
sh.area();
}
}
```

## PROGRAM-05

**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.**

```
import java.util.*;
class Account{
String name;
int accno;
String acc_type;
double balance;
Account(String name,int accno,String acc_type,double balance)
{
this.name=name;
this.accno=accno;
this.acc_type=acc_type;
this.balance=balance;
}
void deposit(double amount)
{
balance+=amount;
}
void withdraw(double amount)
{
if((balance-amount)>=0)
{
balance-=amount;
}
else
{
System.out.println("Insufficient balance! Cannot withdraw");
```

```
}
}
void display()
{
System.out.println("Name:"+" "+name+"Account number:"+"
"+accno+"Account type:"+"
"+acc_type+"Balance:"+" "+balance);
}
}
class Sav_acct extends Account
{
static double rate=5.0;
Sav_acct(String name,int accno,double balance)
{
super(name,accno,"Savings",balance);
}
void interest()
{
balance+=balance*(rate)/100;
System.out.println("Balance:"+balance);
}
}
class Curr_account extends Account
{
private double minBal=500;
private double serviceCharges=50;
Curr_account(String name,int accno,double balance)
```

```
{
super(name,accno,"Current",balance);
}
void checkMin()
{
if (balance<minBal)
{
System.out.println("Balance is less than minimum, penlaty is
imposed"+serviceCharges);
balance-=serviceCharges;
System.out.println("Balance is:"+balance);
}
}
}
class Bank
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("Enter customer name");
String name=s.next();
System.out.println("Enter account number");
int accno=s.nextInt();
System.out.println("Enter the type of account-Current or Savings");
String acc_type=s.next();
System.out.println("Enter the initial balance");
double balance=s.nextDouble();
```

```
Account ob1=new Account(name,accno,acc_type,balance);
Sav_acct sa=new Sav_acct(name,accno,balance);
Curr_account cu=new Curr_account(name,accno,balance);
while (true)
{
if(acc_type.equals("Savings"))
{
System.out.println("Menu");
System.out.println("1.Deposit");
System.out.println("2.Withdraw");
System.out.println("3.Compute Interest for savings account");
System.out.println("4.Display account details");
System.out.println("5.Exit");
System.out.println("Enter your choice:");
int choice=s.nextInt();
switch(choice)
{
case 1:
{
System.out.println("Enter the amount to be deposited");
double amt1=s.nextDouble();
sa.deposit(amt1);
break;
}
case 2:
{
```

```
System.out.println("Enter the amount to be withdrawn");
double amt2=s.nextDouble();
sa.withdraw(amt2);
break;
}
case 3:
{
sa.interest();
break;
}
case 4:
{
sa.display();
break;
}
case 5:
{
break;
}
default:
{
System.out.println("Enter valid input");
break;
}
}
}
```

```
else
{
System.out.println("Menu");
System.out.println("1.Deposit");
System.out.println("2.Withdraw");
System.out.println("3.Display account details");
System.out.println("4.Exit");
System.out.println("Enter your choice:");
int choice=s.nextInt();
switch(choice)
{
case 1:
{
System.out.println("Enter the amount to be deposited");
double amt1=s.nextDouble();
cu.deposit(amt1);
break;
}
case 2:
{
System.out.println("Enter the amount to be withdrawn");
double amt2=s.nextDouble();
cu.withdraw(amt2);
break;
}
case 3:
```

```
{
cu.display();
cu.checkMin();
break;
}
case 4:
{
break;
}
default:
{
System.out.println("Enter valid input");
break;
}
}
}
}
}
}
```

# PROGRAM-06

**Create a package CIE which has two classes- Student and Internals. The class Student has members like usn, name, sem. The class Internals derived from Student 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.**

```
//Student.java
package CIE;
import java.util.Scanner;
public class Student {
protected String usn = new String();
protected String name = new String();
protected int sem;
public void inputStudentDetails()
{
Scanner sc=new Scanner(System.in);
System.out.println(&quot;Enter student usn&quot;);
usn=sc.next();
System.out.println(&quot;Enter student name&quot;);
name=sc.next();
System.out.println(&quot;Enter student semester&quot;);
sem=sc.nextInt();
}
public void displayStudentDetails() {
System.out.println(&quot;student usn:&quot;+ usn);
```

```
System.out.println(&quot;student name:&quot;+name);
System.out.println(&quot; student sem:&quot;+ sem);
}
}
//Internals.java
package CIE;
import java.util.Scanner;
public class Internals extends Student {
protected int marks[] = new int[5];
public void inputCIEmarks()
{
Scanner sc=new Scanner(System.in);
for (int i=0;i&lt;5;i++)
{
System.out.println(&quot;Enter 5 subject marks&quot;);
marks[i]=sc.nextInt();
}
}
}
//Externals.java
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals {
protected int marks[];
protected int finalMarks[];
```

```
public Externals() {
marks = new int[5];
finalMarks = new int[5];
}
public void inputSEEmarks()
{
Scanner sc = new Scanner(System.in);
for(int i=0;i&amp;lt;5;i++)
{
System.out.print(&amp;quot;Subject &amp;quot;+(i+1)+&amp;quot; marks: &amp;quot;);
marks[i] = sc.nextInt();
}
}
public void calculateFinalMarks() {
for(int i=0;i&amp;lt;5;i++)
finalMarks[i] = marks[i]/2 + super.marks[i];
}
public void displayFinalMarks() {
displayStudentDetails();
for(int i=0;i&amp;lt;5;i++)
System.out.println(&amp;quot;Subject &amp;quot; + (i+1) + &amp;quot;: &amp;quot; +
finalMarks[i]);
}
}
//Main1.java
import SEE.Externals;
class Main1 {
```

```
public static void main(String args[])
{
int numOfStudents = 2;
Externals finalMarks[] = new
Externals[numOfStudents];
for(int i=0;i&amp;lt;numOfStudents;i++)
{
finalMarks[i] = new Externals();
finalMarks[i].inputStudentDetails();
System.out.println(&amp;quot;Enter CIE marks&amp;quot;);
finalMarks[i].inputCIEmarks();
System.out.println(&amp;quot;Enter SEE marks&amp;quot;);
finalMarks[i].inputSEEmarks();
}
System.out.println(&amp;quot;Displaying data:\n&amp;quot;);
for(int i=0;i&amp;lt;numOfStudents;i++)
{
finalMarks[i].calculateFinalMarks();
finalMarks[i].displayFinalMarks();
} //end of for loop
} // end of public main
} //end of class main
```

# PROGRAM-07

**Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called “Father” and derived class called “Son” which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son’s age and throws an exception if son’s age**

is >=father’s age.

```
import java.util.*;
class WrongAge extends Exception
{
public WrongAge(String s)
{
super(s);
}
}
class Father
{
Scanner sc=new Scanner(System.in);
public int F_age;
public Father() throws WrongAge
{
System.out.println("Enter Father's age");
F_age=sc.nextInt();
if (F_age<0)
{
throw new WrongAge("Age cannot be negative");
}
```

```
}
public void display()
{
System.out.println("Father's age="+F_age);
}
}
class Son extends Father
{
private int S_age;
public Son() throws WrongAge
{
System.out.println("Enter son's age");
S_age=sc.nextInt();
if (S_age>F_age)
{
throw new WrongAge("Son's age cannot be greater than father's age");
}
else if (S_age<0)
{
throw new WrongAge("Age cannot be negative");
}
else if (S_age==F_age)
{
throw new WrongAge("Age cannot be same");
}
else
```

```
{
System.out.println("Valid age");
}
}
public void display()
{
System.out.println("Father's age="+F_age);
System.out.println("Son's age="+S_age);
}
}
class Lab7
{
public static void main(String args[])
{
try
{
Son son=new Son();
son.display();
}
catch(WrongAge e)
{
System.out.println("Exception caught");
System.out.println(e.getMessage());
}
}
}
```

## PROGRAM-08

**Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.**

```
class BMS extends Thread{
public void run(){
for(int i=1;i<=5;i++){
System.out.println("bms college of engineering"+i);
try
{
Thread.sleep(10000);
}
catch(InterruptedException e){
e.printStackTrace();
}
}
}
}
class CSE extends Thread{
public void run(){
for(int i=1;i<=10;i++){
System.out.println("Cse"+i);
try
{
Thread.sleep(2000);
}
catch(InterruptedException e){
```

```
e.printStackTrace();
}
}
}
}
class threadMain{
public static void main(String a[]){
BMS obj1=new BMS();
CSE obj2=new CSE();
obj1.start();
obj2.start();
}
}
```

## PROGRAM-09

**Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.**

```
import java.awt.*;
import java.awt.event.*;
public class DivisionMain1 extends Frame implements ActionListener
{
TextField num1,num2;
```

```
Button dResult;
Label outResult;
String out="";
double resultNum;
int flag=0;
public DivisionMain1()
{
setLayout(new FlowLayout());
dResult = new Button("RESULT");
Label number1 = new Label("Number 1:",Label.RIGHT);
Label number2 = new Label("Number 2:",Label.RIGHT);
num1=new TextField(5);
num2=new TextField(5);
outResult = new Label("Result:",Label.RIGHT);
add(number1);
add(num1);
add(number2);
add(num2);
add(dResult);
add(outResult);
num1.addActionListener(this);
num2.addActionListener(this);
dResult.addActionListener(this);
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent we)
```

```
{
System.exit(0);
}
});
}
public void actionPerformed(ActionEvent ae)
{
int n1,n2;
try
{
if (ae.getSource() == dResult)
{
n1=Integer.parseInt(num1.getText());
n2=Integer.parseInt(num2.getText());
/*if(n2==0)
throw new ArithmeticException();*/
out=n1+" "+n2;
resultNum=n1/n2;
out+=String.valueOf(resultNum);
repaint();
}
}
catch(NumberFormatException e1)
{
flag=1;
out="Number Format Exception! "+e1;
```

```
repaint();
}
catch(ArithmeticException e2)
{
flag=1;
out="Divide by 0 Exception! "+e2;
repaint();
}
}
public void paint(Graphics g)
{
if(flag==0)
g.drawString(out,outResult.getX()+outResult.getWidth(),outResult.getY()+outR
esult.getHeight()-8);
else
g.drawString(out,100,200);
flag=0;
}
public static void main(String[] args)
{
DivisionMain1 dm=new DivisionMain1();
dm.setSize(new Dimension(800,400));
dm.setTitle("DivionOfIntegers");
dm.setVisible(true);
}
}
```

# PROGRAM-10

**10.A)Demonstrate Inter process Communication and deadlock**

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

```
System.out.println("InterruptedException caught");
}
}
this.n = n;
valueSet = true;
System.out.println("Put: " + n);
System.out.println("Intimate Consumer\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<5) {
q.put(i++);
}
}
}
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<5) {
int r=q.get();
System.out.println("consumed:"+r);
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.");
}
}
```

**10B)DEADLOCK**

```
class A {
synchronized void foo(B b) {
String name =Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
```

```
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
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("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
a.last();
}
void last() {
System.out.println("Inside A.last");
```

```
}
}
class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b); // get lock on a in this thread.
System.out.println("Back in main thread");
}
public void run() {
b.bar(a); // get lock on b in other thread.
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```