

**VISVESVARAYA TECHNOLOGICAL
UNIVERSITY**

“JnanaSangama”, Belgaum -590014, Karnataka.



LAB REPORT
on
Object Oriented Java Programming
(23CS3PCOOJ)

Submitted by

Hitha Harish (1BM23CS115)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



B.M.S. COLLEGE OF ENGINEERING

(Autonomous Institution under VTU)

BENGALURU-560019

Sep-202

B.M.S. College of Engineering,
Bull Temple Road, Bangalore 560019
(Affiliated To Visvesvaraya Technological University, Belgaum)
Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the Lab work entitled “Object Oriented Java(23CS3PCOOJ)” carried out by **Hitha Harish (1BM23CS115)**, who is bonafide student of **B.M.S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum. The Lab report has been approved as it satisfies the academic requirements in respect of an Object Oriented Java (23CS3PCOOJ) work prescribed for the said degree.

Ambuja K Assistant Professor Department of CSE, BMSCE	Dr. Kavitha Sooda Professor & HOD Department of CSE, BMSCE
-------------------------------------------------------------	------------------------------------------------------------------

Index

Sl. No.	Date	Experiment Title	Page No.
1	24/9/24	Programs in C	4
2	1/10/24	Addition, Product of 2 numbers. Quadratic Expressions	14
3	8/10/24	Class Student to take details of the student and calculate SGPA	18
4	15/10/24	Class Book to take the details of a book using Constructors, and set methods. Display the details using get methods and <code>toString()</code>	25
5	22/10/24	Abstract class Animal Abstract class Shape	31
6	29/10/24	Class Account – Savings and Current	39
7	10/11/24	Package CIE and SEE	47
8	19/11/24	Interface Programs	55
9	26/11/24	Exceptions	60
10	3/12/24	Threads	64

Github Link: <https://github.com/HithaHarish-csbmsce/OOJ>

Lab 1

Program1:

Code:

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    if (num % 2 == 0) {
        printf("%d is even.\n", num); // If remainder is 0, it's even
    } else {
        printf("%d is odd.\n", num); // Otherwise, it's odd
    }
    return 0;
}
```

Output:

Enter an integer: 7
7 is odd.

Enter an integer: 12
12 is even.

Program2:

```
#include <stdio.h>
int main() {
    int num;
    printf("Enter an integer: ");
    scanf("%d", &num);
    if (num % 2 != 0) {
        int square = num * num; // Calculate the square of the odd number
        printf("The square of %d is %d.\n", num, square);
        if (square % 2 == 0) {
            printf("The square of %d is even.\n", num);
        } else {
            printf("The square of %d is odd.\n", num);
        }
    } else {
        printf("%d is not an odd number.\n", num);
    }
    return 0;
}
```

```
}
```

Output:

```
Enter an integer: 5
The square of 5 is 25.
The square of 5 is odd.
```

```
Enter an integer: 3
The square of 3 is 9.
The square of 3 is odd.
```

```
Enter an integer: 4
4 is not an odd number.
```

Program3:

```
#include <stdio.h>
int main() {
    int numerator, denominator;
    printf("Enter numerator: ");
    scanf("%d", &numerator);
    printf("Enter denominator: ");
    scanf("%d", &denominator);
    if (denominator == 0) {
        printf("Error: Division by zero is not allowed!\n");
    } else {
        int result = numerator / denominator;
        printf("Result: %d / %d = %d\n", numerator, denominator, result);
    }
    return 0;
}
```

Output:

```
Enter numerator: 10
Enter denominator: 2
Result: 10 / 2 = 5
Enter numerator: 10
Enter denominator: 0
Error: Division by zero is not allowed!
```

Program4:

```
#include <stdio.h>
```

```

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);
    if (num % 2 == 0) {
        printf("%d is even.\n", num);
    } else {
        printf("%d is odd.\n", num);
    }
    printf("Numbers from 1 to %d in ascending order:\n", num);
    int i = 1;
    while (i <= num) {
        printf("%d ", i);
        i++;
    }
    printf("\n");
    printf("Numbers from %d down to 0 in descending order:\n", num);
    int j = num;
    do {
        printf("%d ", j);
        j--;
    } while (j >= 0);
    printf("\n");

    return 0;
}

```

Output:

Enter a number: 5

5 is odd.

Numbers from 1 to 5 in ascending order:

1 2 3 4 5

Numbers from 5 down to 0 in descending order:

5 4 3 2 1 0

Enter a number: 6

6 is even.

Numbers from 1 to 6 in ascending order:

1 2 3 4 5 6

Numbers from 6 down to 0 in descending order:

6 5 4 3 2 1 0

Program5:

```
#include <stdio.h>
```

```

int main() {
    int num1, num2;
```

```

char operator;
printf("Enter first number: ");
scanf("%d", &num1);
printf("Enter an operator (+, -, *, /, %): ");
scanf(" %c", &operator);
printf("Enter second number: ");
scanf("%d", &num2);

switch (operator) {
    case '+':
        printf("%d + %d = %d\n", num1, num2, num1 + num2);
        break;
    case '-':
        printf("%d - %d = %d\n", num1, num2, num1 - num2);
        break;
    case '*':
        printf("%d * %d = %d\n", num1, num2, num1 * num2);
        break;
    case '/':
        if (num2 == 0) {
            printf("Error: Division by zero is not allowed!\n");
        } else {
            printf("%d / %d = %.2f\n", num1, num2, (float)num1 / num2);
        }
        break;
    case '%':
        // Check for modulus by zero
        if (num2 == 0) {
            printf("Error: Modulus by zero is not allowed!\n");
        } else {
            printf("%d %% %d = %d\n", num1, num2, num1 % num2);
        }
        break;
    default:
        printf("Error: Invalid operator!\n");
        break;
}

return 0;
}

```

Output:

```

Enter first number: 12
Enter an operator (+, -, *, /, %): +
Enter second number: 5
12 + 5 = 17

```

i) Check Even or Odd

```

• #include <stdio.h>
int main()
{
    int n;
    printf ("Enter the number: ");
    scanf ("%d", &n);
    if (n % 2 == 0)
    {
        printf ("The number %d is even");
    }
    else
    {
        printf ("The number %d is odd");
    }
}

```

OUTPUT:

Enter the number: 7
The number 7 is odd

```

• #include <stdio.h>
int main()
{
    int n;
    printf ("Enter the number: ");
    scanf ("%d", &n);
    if ((n / 2) * 2 == n)
    {
        printf ("Even");
    }
}

```

```
else  
{  
    printf ("\\nOdd")  
}  
}
```

OUTPUT:

Enter the number : 2

Even

Enter the number: 3

Odd

2) Square of Odd Numbers

```
#include<stdio.h>  
int main ()  
{  
    int num, sq, i = 0;  
    printf ("Enter the number: ");  
    scanf ("%d", &num);  
    for (i=0; i<=num; i++)  
    {  
        if (i%2 == 1)  
        {  
            sq = i * i;  
            printf ("\n%d", sq);  
        }  
    }  
}
```

OUTPUT:

Enter the number: 6

1
9
25

3) Divide by zero.

```
#include <stdio.h>
int main ()
{
    int num;
    printf ("Enter the number: ");
    scanf ("%d", &num);
    printf ("%d", num/0);
}
```

OUTPUT:

Exception has occurred.
Arithmetic exception

~~Enter the number: 6~~

4) Program to use if, else, while, do while

```
#include <stdio.h>
int main ()
{
    int num, i;
    printf ("Enter a number: ");
```

Page No. _____
Date. _____

```

scanf (" %d", &num);
if (num % 2 == 0)
{
    printf (" The number is even");
}
else
{
    printf (" The number is odd");
}
printf ("\\nAscending order");
while (i < num)
{
    printf (" \\n%d", i);
    i++;
}
printf ("\\nDescending order");
do
{
    i--;
    printf (" \\n%d", i);
} while (i > 0);
}

```

OUTPUT

Enter a number: 7

The number is odd

Ascending order

0

1

2

3

4

5

6

7

Descending order

7

6

5

4

3

2

1

0

5) Switch case calculator

```
#include <stdio.h>
int main()
{
    char operator;
    float n1, n2, r=0;
    printf ("Enter operator:");
    scanf ("%c", &operator);
    printf ("Enter 2 numbers");
    scanf ("%f %f", &n1, &n2);
    switch (operator)
    {
        case '+':
            r = n1+n2; break;
        case '-':
            r = n1-n2; break;
        case '*':
            r = n1*n2; break;
        case '/':
            r = n1/n2; break;
    }
}
```

default:

```
    printf ("Default")  
} printf ("\nThe result is %d ", n); }
```

O/P:

Enter operator: +

Enter 2 numbers: 2 3

The result is 5

6) PRO's of C

- easy to learn
- fast.
- Dynamic memory allocation

CON's of C

- OOPs concepts are absent.
- Manual memory management
- lack of built in safety.

X 2019

Lab2

Program1:

```
public class SimpleCalculator {  
    public static void main(String[] args) {  
        int num1 = 10;  
        int num2 = 5;  
        int sum = num1 + num2;  
        int difference = num1 - num2;  
        System.out.println("Sum of " + num1 + " and " + num2 + " is: " + sum);  
        System.out.println("Difference between " + num1 + " and " + num2 + " is: " + difference);  
    }  
}
```

Output:

Sum of 10 and 5 is: 15
Difference between 10 and 5 is: 5

Program2:

```
import java.util.Scanner;  
public class Calculator {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the first number: ");  
        double num1 = scanner.nextDouble();  
        System.out.print("Enter the second number: ");  
        double num2 = scanner.nextDouble();  
        double product = num1 * num2;  
        double quotient = 0.0;  
        if (num2 != 0) {  
            quotient = num1 / num2;  
            System.out.println("Division result: " + quotient);  
        } else {  
            System.out.println("Error: Division by zero is not allowed.");  
        }  
        System.out.println("Product (multiplication) of " + num1 + " and " + num2 + " is: " + product);  
    }  
}
```

Output:

Enter the first number: 10
Enter the second number: 5
Product (multiplication) of 10.0 and 5.0 is: 50.0
Division result: 2.0

Program3:

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

public class QuadraticEquation {

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

        System.out.print("Enter coefficient a: ");
        double a = scanner.nextDouble();

        System.out.print("Enter coefficient b: ");
        double b = scanner.nextDouble();

        System.out.print("Enter coefficient c: ");
        double c = scanner.nextDouble();

        double discriminant = b * b - 4 * a * c;

        if (discriminant > 0) {
            double root1 = (-b + Math.sqrt(discriminant)) / (2 * a);
            double root2 = (-b - Math.sqrt(discriminant)) / (2 * a);
            System.out.println("Real and distinct roots: " + root1 + " and " + root2);
        } else if (discriminant == 0) {
            double root = -b / (2 * a);
            System.out.println("Real and identical roots: " + root);
        } else {
            System.out.println("No real roots");
        }
    }
}
```

Output:

```
Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 2
Real and distinct roots: 2.0 and 1.0
```

1) ADD / SUBTRACT 2 NUMBERS

```
import java.util.*;
public class Main {
    public static void main (String [] args) {
        int x, y; x=5, y=6;
        System.out.println ("Addition of x and
y is " + (x+y));
        System.out.println ("Subtraction of x and
y is " + (x-y));
    }
}
```

OUTPUT:

Addition of x and y is 11
Subtraction of x and y is -1

2) calculate the product of 2 Numbers by taking user input.

~~Not Done~~

```
import java.util.*;
public class Main {
    public static void main (String [] args) {
        Scanner inp = new Scanner (System.in);
        int a = inp.nextInt ();
        int b = inp.nextInt ();
        int product = a * b;
        System.out.println ("The product
is " + product);
    }
}
```

}

```
else if ((b*b - 4*a*c) > 0) {  
    int D = b*b - 4*a*c;  
    double x1 = (-b + sqrt(D)) / (2*a);  
    double x2 = (-b - sqrt(D)) / (2*a);  
    System.out.println ("The roots are  
real and distinct");  
    System.out.println ("x1 and x2  
are " + x1 + ", " + x2);  
}
```

else {

```
    System.out.println ("The roots are  
not real");  
}
```

{

if (0.0){}

}

OUTPUT:

Enter a: 5

Enter b : 7

Enter c : 2

The roots are real and distinct.

x1 and x2 are -0.4 , -1.

Lab3

Program1:

```
import java.util.Scanner;

class Subject {
    int subjectMarks;
    int credits;
    int grade;

    public void calculateGrade() {
        if (subjectMarks < 40)
            grade = 0;
        else if (subjectMarks > 100)
            grade = 4; // Invalid grade, assuming it's an error
        else {
            if (subjectMarks >= 90)
                grade = 10;
            else if (subjectMarks >= 80)
                grade = 9;
            else if (subjectMarks >= 70)
                grade = 8;
            else if (subjectMarks >= 60)
                grade = 7;
            else if (subjectMarks >= 50)
                grade = 6;
            else if (subjectMarks >= 40)
                grade = 5;
            else
                grade = -1;
        }
    }
}

class Student1 {
    String name;
    String usn;
    double SGPA;
    Subject subject[];
    Scanner s;
    Student1() {
        subject = new Subject[8];
        for (int i = 0; i < 8; i++) {
            subject[i] = new Subject();
        }
        s = new Scanner(System.in);
    }
}
```

```

public void getStudentDetails() {
    System.out.print("Enter student name: ");
    this.name = s.nextLine();
    System.out.print("Enter student USN: ");
    this.usn = s.nextLine();
}

public void getMarks() {
    for (int i = 0; i < 8; i++) {
        System.out.print("Enter marks for subject " + (i + 1) + ": ");
        subject[i].subjectMarks = s.nextInt();
        System.out.print("Enter credits for subject " + (i + 1) + ": ");
        subject[i].credits = s.nextInt();
        subject[i].calculateGrade();
    }
    s.nextLine();
}

public void computeSGPA() {
    double totalPoints = 0;
    int totalCredits = 0;

    for (int i = 0; i < 8; i++) {
        totalPoints += subject[i].grade * subject[i].credits;
        totalCredits += subject[i].credits;
    }

    SGPA = (totalCredits == 0) ? 0 : totalPoints / totalCredits;
}

public void displayResults() {
    System.out.println("Student Name: " + name);
    System.out.println("USN: " + usn);
    System.out.printf("SGPA: %.2f%n", SGPA);
}

public class Student {
    public static void main(String[] args) {
        Student1 s1 = new Student1();
        s1.getStudentDetails();
        s1.getMarks();
        s1.computeSGPA();
        s1.displayResults();

    }
}

```

Output:

Enter student name: Hitha Harish

Enter student USN: 115

Enter marks for subject 1: 90

Enter credits for subject 1: 4

Enter marks for subject 2: 89

Enter credits for subject 2: 4

Enter marks for subject 3: 98

Enter credits for subject 3: 4

Enter marks for subject 4: 86

Enter credits for subject 4: 3

Enter marks for subject 5: 87

Enter credits for subject 5: 3

Enter marks for subject 6: 90

Enter credits for subject 6: 2

Enter marks for subject 7: 97

Enter credits for subject 7: 1

Enter marks for subject 8: 94

Enter credits for subject 8: 1

Student Name: Hitha Harish

USN: 115

SGPA: 9.55

14 Develop a program to create a class Student with members USN, Name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;  
  
class Subject {  
    int subjectMarks;  
    int credits;  
    int grade;  
  
    public void calculateGrade() {  
        if (subjectMarks < 40) {  
            grade = 0;  
        } else if (subjectMarks > 100) {  
            grade = 4; // Invalid, error.  
        } else {  
            if (subjectMarks >= 90)  
                grade = 10;  
            else if (subjectMarks >= 80)  
                grade = 9;  
            else if (subjectMarks >= 70)  
                grade = 8;  
            else if (subjectMarks >= 60)  
                grade = 7;  
            else if (subjectMarks >= 50)  
                grade = 6;  
            else if (subjectMarks >= 40)  
                grade = 5;  
            else  
                grade = -1;  
        }  
    }  
}
```

```
class Student1 {  
    String name;  
    String usn;  
    double SGPA;  
    Subject subject[];  
    Scanner s;  
    Student1 () {  
        subject = new Subject[8];  
        for (int i=0; i<8; i++) {  
            subject[i] = new Subject();  
        }  
        s = new Scanner (System.in);  
    }
```

```
public void getStudentDetails() {  
    System.out.println("Enter student name");  
    this.name = s.nextLine();  
    System.out.println("Enter USN:");  
    this.usn = s.nextLine();  
}
```

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

```
Page No. _____  
Date. _____ / _____ / _____
```

```
        subject[i].calculateGrade();  
    }  
    s.nextLine();  
}
```

```
public void computeSGPA(){  
    double totalPoints = 0;  
    int totalCredits = 0;  
    for(int i=0; i<8; i++){  
        totalPoints += subject[i].grade * subject[i].credits;  
        totalCredits += subject[i].credits;  
    }  
    SGPA = (totalCredits == 0) ? 0 : totalPoints/  
        totalCredits;
```

```
public void displayResults(){  
    System.out.println("Student Name: " + name);  
    System.out.println("USN: " + um);  
    System.out.println("SGPA: " + SGPA);  
}
```

```
public class Student{  
    public static void main(String[] args){  
        Student1 s1 = new Student1();  
        s1.getStudentDetails();  
        s1.getMaths();  
        s1.computeSGPA();  
        s1.displayResults();  
    }  
}
```

OUTPUT:

Enter student name: Hitha Haresh

Enter student USN: 1BM23CS115

Enter marks for subject 1: 98

Enter credits for subject 1: 4

Enter marks for subject 2: 89

Enter credits for subject 2: 4

Enter marks for subject 3: 78

Enter credits for subject 3: 4

(i) credits;

Enter marks for subject 4: 98

Enter credits for subject 4: 3

Enter marks for subject 5: 98

Enter credits for subject 5: 3

Enter marks for subject 6: 78

Enter credits for subject 6: 2

Enter marks for subject 7: 89

Enter credits for subject 7: 1

Enter marks for subject 8: 78

Enter credits for subject 8: 1

Student Name: Hitha Haresh

USN: 1BM23CS115

SGPA: 9.14

~~for
for~~

Lab4:

Program1:

```
import java.util.*;  
class Book {  
    public String name;  
    public String author;  
    public double price;  
    public int num_pages;  
  
    public Book(String name, String author, double price, int num_pages) {  
        this.name = name;  
        this.author = author;  
        this.price = price;  
        this.num_pages = num_pages;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getAuthor() {  
        return author;  
    }  
  
    public void setAuthor(String author) {  
        this.author = author;  
    }  
  
    public double getPrice() {  
        return price;  
    }  
  
    public void setPrice(double price) {  
        this.price = price;  
    }  
  
    public int getNumPages() {  
        return num_pages;  
    }  
  
    public void setNumPages(int num_pages) {  
        this.num_pages = num_pages;  
    }
```

```

        }

    @Override
    public String toString() {
        return "Book Name: " + name + ", Author: " + author +
               ", Price: $" + price + ", Pages: " + num_pages;
    }
}

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

        System.out.print("Enter the number of books: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline

        Book[] books = new Book[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter details for Book " + (i + 1) + ":");
            System.out.print("Name: ");
            String name = scanner.nextLine();

            System.out.print("Author: ");
            String author = scanner.nextLine();

            System.out.print("Price: ");
            double price = scanner.nextDouble();

            System.out.print("Number of Pages: ");
            int num_pages = scanner.nextInt();
            scanner.nextLine(); // Consume the newline

            books[i] = new Book(name, author, price, num_pages);
        }
        System.out.println("\nDetails of all books:");
        for (Book book : books) {
            System.out.println(book);
        }
        scanner.close();
    }
}

```

Output:

Enter the number of books: 2
 Enter details for Book 1:

Name: abc

Author: xyz

Price: 50

Number of Pages: 45

Enter details for Book 2:

Name: pqr

Author: lmn

Price: 78

Number of Pages: 96

Details of all books:

Book Name: abc, Author: xyz, Price: \$50.0, Pages: 45

Book Name: pqr, Author: lmn, Price: \$78.0, Pages: 96

- 1) Create a class Book which contains 4 members : name, author, price, num-page. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() that could display the complete details of the book. Develop a Java program to create n book objects.

```

import java.util.*;
class Book {
    public String Name;
    public String Author;
    public int Price;
    public int num-page;

    public Book (String Name, String Author,
                int Price, int num-page) {
        this.Name = Name;
        this.Author = Author;
        this.Price = Price;
        this.num-page = num-page;
    }

    public String getName () {
        return Name;
    }

    public void setName (String Name) {
        this.Name = Name;
    }

    public String Author () {
        return Author;
    }
}
  
```

Page No. _____
Date: ____ / ____ / ____

```
public void setAuthor ( String Author ) {  
    this.Author = Author;  
}  
public int getPrice () {  
    return Price;  
}  
public void setPrice ( int Price ) {  
    this.Price = Price;  
}  
public int getnumPages () {  
    return numPages;  
}  
public void setnumPages ( int numPages ) {  
    this.numPages = numPages;  
}
```

@Overrides

```
public String toString () {  
    return "Book Name:" + Name + "Author:" +  
        Author + "Price" + Price + "No. of Pages" + numPages;  
}
```

```
public class Main {
```

```
    public static void main ( String [] args ) {  
        Scanner scanner = new Scanner ( System.in );  
        System.out.println ("Enter no. of books:");  
        int n = scanner.nextInt ();  
        scanner.nextLine ();  
        Book [] books = new Book [n];  
        for ( int i = 0 ; i < n ; i++ ) {  
            System.out.println ("Enter the details:");  
            System.out.print ("Name: ");
```

Page No. _____
Date: _____

```
String Name = scanner.nextLine();
System.out.println("Author: ");
String Author = scanner.nextLine();
System.out.println("Price: ");
float Price = scanner.nextLine();
System.out.println("Number of pages");
int numPages = scanner.nextInt();
books[i] = new Book (Name, Author, Price,
    numPages);
```

```
System.out.println("Details of books:");
for (Book book : books) {
    System.out.println(book);
}
```

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

```
}
```

OUTPUT

Enter the number of books: 2

Enter the details:

Name: xyz

Author: abc

Price: 30

Number of Pages: 25

Details of books

BOOK Name: xyz, Author: abc, Price: 30, Number of
Pages: 25

Lab5:

Program1:

```
import java.util.*;
abstract class Animal{
String Name;
String Food;
String Sleep;
abstract void eat();
abstract void sleep();
}
class Lion extends Animal{
    void eat(){
        System.out.println("Carnivorous");
    }
    void sleep(){
        System.out.println("12 hours");
    }
}
class Tiger extends Animal{
    void eat(){
        System.out.println("Carnivorous");
    }
    void sleep(){
        System.out.println("10 hours");
    }
}
class Deer extends Animal{
    void eat(){
        System.out.println("Herbivorous");
    }
    void sleep(){
        System.out.println("8 hours");
    }
}
public class Main{
    public static void main(String[] args){
        Lion L=new Lion();
        L.eat();
        L.sleep();
        Tiger T=new Tiger();
        T.eat();
        T.sleep();
        Deer D=new Deer();
        D.eat();
        D.sleep();
    }
}
```

}

Output:

Carnivorous
12 hours
Carnivorous
10 hours
Herbivorous
8 hours

Program2:

```
import java.util.*;  
abstract class Shape{  
    abstract void printArea();  
}  
class Rectangle extends Shape{  
    double l;  
    double b;  
    Rectangle(double l,double b){  
        this.l=l;  
        this.b=b;  
    }  
    void printArea(){  
        System.out.println(l*b);  
    }  
}  
class Triangle extends Shape{  
    double h;  
    double b;  
    Triangle(double h,double b){  
        this.h=h;  
        this.b=b;  
    }  
    void printArea(){  
        System.out.println(0.5*h*b);  
    }  
}  
class Circle extends Shape{  
    double r;  
    Circle(double r){  
        this.r=r;  
    }  
    void printArea(){  
        System.out.println(3.14*r*r);  
    }  
}  
public class Main1{
```

```
public static void main(String[] args){  
    Rectangle R=new Rectangle(5,6);  
    R.printArea();  
    Triangle T=new Triangle(2,5);  
    T.printArea();  
    Circle C=new Circle(1);  
    C.printArea();  
  
}
```

Output:

30.0
5.0
3.14

①

LAB - 5

```
import java.util.*;  
abstract class Animal {  
    String Name;  
    String Food;  
    String Sleep;  
    Animal (String Name, String Food,  
            String Sleep)  
    {  
        this.Name = Name;  
        this.Food = Food;  
        this.Sleep = Sleep;  
    }  
}
```

abstract void eat();

pass

abstract void sleep();

pass

```
class Lion extends Animal {
```

```
    void eat (String Food) {  
        System.out.println ("Carnivore");  
    }  
}
```

```
    void sleep () {  
        System.out.println ("12 Hours");  
    }  
}
```

class Tiger extends Animal {

void eat () {
 System.out.println ("Carnivore");
}

void sleep () {
 System.out.println ("10 hours");
}

class Deer extends Animal {

void eat () {
 System.out.println ("Herbivore");
}

void sleep () {
 System.out.println ("8 Hours");
}

public class Main {

 public static void main (String [] args) {

 Animal A = new Animal();

 A.eat();

 Lion Lion1 = new Lion();

 Lion1.eat();

 Lion1.sleep();

 Deer Deer1 = new Deer();

Deer1.eat();

Deer1.sleep();

Tiger1.Tiger1 = new Tiger();

Tiger1.eat();

Tiger1.sleep();

}

~~for~~ 1..0

OUTPUT:

carnivorous

12 hours

Herbivorous

8 Hours

carnivorous

10 hours.

④

1

② Develop a java program to create an abstract class named Shape that contains 2 integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle, circle such that each of these contain only the method printArea() & extends the class.

import java.util.Scanner;

abstract class Shape {

 abstract printArea();

}

class Rectangle extends Shape {

 Rectangle (double l, double b) {

 double l;

 this.l = l;

 double b;

 this.b = b;

}

 void printArea () {

 System.out.println (l * b);

}

class Triangle extends Shape {

 Triangle (double b, double h) {

 double b;

 double h;

 this.b = b;

 this.h = h;

}

 void printArea () {

66
Page No.
Date: / / 99

```

        system.out.println(0.5 * b * h);
    }

class Circle extends Shape {
    Circle() {
        double radius;
        void circle(double radius) {
            this.radius = radius;
        }
        void printArea() {
            System.out.println(3.14 * radius
                * radius);
        }
    }

public class Main1 {
    public static void main(String[] args) {
        Triangle T = new Triangle(2, 5);
        T.printArea();
        Rectangle R = new Rectangle(5, 6);
        R.printArea();
        Circle C = new Circle(2);
        C.printArea();
    }
}

OUTPUT:
5.0
30.0
3.14

```

Lab6:

Program1:

```
class Account {  
    String customerName;  
    int accountNumber;  
    String accountType;  
    double balance;  
  
    public Account(String customerName, int accountNumber, String accountType, double  
initialBalance) {  
        this.customerName = customerName;  
        this.accountNumber = accountNumber;  
        this.accountType = accountType;  
        this.balance = initialBalance; // Corrected to use initialBalance  
    }  
  
    public void deposit(double amount) {  
        if (amount > 0) {  
            this.balance += amount; // Simplified  
            System.out.println("Deposited: $" + amount);  
        } else {  
            System.out.println("Invalid deposit amount.");  
        }  
    }  
  
    public void displayBalance() {  
        System.out.println("Account Balance: $" + balance);  
    }  
  
    public void withdraw(double amount) {  
        if (amount <= balance) {  
            balance -= amount;  
            System.out.println("Withdrew: $" + amount);  
        } else {  
            System.out.println("Insufficient balance.");  
        }  
    }  
}  
  
class SavAcct extends Account {  
    double interestRate;  
  
    public SavAcct(String customerName, int 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 added: $" + interest);
}
}

class CurAcct extends Account {
    double minimumBalance;
    double serviceCharge;

    public CurAcct(String customerName, int accountNumber, double balance, double
minimumBalance, double serviceCharge) {
        super(customerName, accountNumber, "Current", balance);
        this.minimumBalance = minimumBalance;
        this.serviceCharge = serviceCharge;
    }

    @Override
    public void withdraw(double amount) {
        if (amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: $" + amount);
            if (balance < minimumBalance) {
                balance -= serviceCharge;
                System.out.println("Service charge imposed: $" + serviceCharge);
            }
        } else {
            System.out.println("Insufficient balance.");
        }
    }
}

public class Bank {
    public static void main(String[] args) {
        SavAcct S = new SavAcct("Hitha Harish", 2152, 10000, 5);
        S.computeInterest();
        S.deposit(1000);
        S.displayBalance();
        S.withdraw(500);
        S.displayBalance();

        System.out.println();

        CurAcct C = new CurAcct("Jane", 102, 3000, 1000, 50);
        C.deposit(2000);
        C.displayBalance();
        C.withdraw(2000);
        C.withdraw(2500);
    }
}

```

```
        C.displayBalance();  
    }  
}
```

Output:

```
Interest added: $500.0  
Deposited: $1000.0  
Account Balance: $11500.0  
Withdrew: $500.0  
Account Balance: $11000.0
```

```
Deposited: $2000.0  
Account Balance: $5000.0  
Withdrew: $2000.0  
Withdrew: $2500.0  
Service charge imposed: $50.0  
Account Balance: $450.0
```

CODE :

```
import java.util.*;  
  
class Account {  
    String customername;  
    int accountnumber;  
    String Typeofacc;  
    double balance;  
    public Account (String customername, int  
                    accountnumber, String Typeof acc,  
                    double balance)  
    {  
        this.customername = customername;  
        this.accountnumber = accountnumber;  
        this.Typeofacc = Typeofacc;  
        this.balance = balance;  
    }  
    public void deposit (double amount)  
    {  
        if (amount > 0)  
        {  
            balance += amount;  
        }  
    }  
}
```

```
        } system.out.println ("Deposited : $" +  
        amount);  
    else {  
        system.out.println ("Invalid Deposit");  
    }
```

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

```
public void withdraw ($ double amount) {  
    balance -= amount;  
    system.out.println ("The balance is now  
    $" + balance);  
}
```

```
class SavingsAccount extends Account {  
    double interestRate;
```

```
    public SavingsAccount (String customer  
    -username, int accountNumber, double balance,  
    double interestRate) {  
        super (customerName, accountNumber,  
        "Savings", balance);  
        this.interestRate = interestRate;  
    }
```

```
    public void computeInterest () {  
        double interest = (balance * interestRate) / 100;  
        balance += interest;
```

Page No. _____
Date. _____

System.out.println ("Interest added \$" + interest);

}

@Override

public void

}

class Current extends Account {

double min^{imum}Balance;

double serviceCharge;

Current

public Account (String customerName,
int accountNumber, double balance,
double initialBalance, double service
-charge) ;

{

super (customerName, accountNumber,
"current", balance);

this.minimumBalance = minimumBalance;

this.serviceCharge = serviceCharge;

@Override

public void withdraw (double amount) {

if (balance >= amount) {

balance -= amount;

System.out.println ("The balance

if (balance < minimumBalance) {

balance -= serviceCharge;

System.out.println ("Service charge
levied");

100;

System.out.println ("Balance is: \$" +
balance);

Page No. _____
Date. _____

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

```
        Account A = new Account ();
```

```
        SavingsAccount S = new SavingsAccount (  
            "Hithatlaish", 2152, 10000, 5)
```

```
        S.computeInterest ();
```

```
        S.deposit(1000);
```

```
        S.displayBalance ();
```

```
        S.withdraw(500);
```

```
        S.displayBalance ();
```

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

```
        CurrentAccount C = new CurrentAccount (
```

```
            "Jane", 102, 3000, 1000, 50);
```

```
        C.deposit(2000)
```

```
        C.displayBalance ();
```

```
        C.withdraw(2000);
```

```
        C.withdraw(2500);
```

```
        C.displayBalance ();
```

```
}
```

~~Output~~
 OUTPUT

Interest added : \$ 000.500.0

Deposited : \$ 1000.0

Account Balance : \$ 11500.0

Withdrawn : \$ 500.0

Account Balance : \$ 11000.0

Deposited : \$ 2000.0

Account Balance : \$ 13000.0

withdrawn: \$ 2000.0

withdrawn: \$ 2500.0

Service charge imposed: \$ 50.0

Current Balance: \$ 450.0

Lab7:

Program1:

```
package CIE;

import java.util.Scanner;

public class Student {
    protected String usn;
    protected String name;
    protected int sem;

    // Method to input student details
    public void inputStudentDetails() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter USN: ");
        usn = s.nextLine();
        System.out.println("Enter Name: ");
        name = s.nextLine();
        System.out.println("Enter Semester: ");
        sem = s.nextInt();
    }

    // Method to display student details
    public void displayStudentDetails() {
        System.out.println("USN: " + usn);
        System.out.println("Name: " + name);
        System.out.println("Semester: " + sem);
    }
}

package CIE;

import java.util.Scanner;

public class Internals extends Student {
    protected int[] marks = new int[5]; // Array to store internal marks for 5 courses

    // Method to input internal marks for five courses
    public void inputCIEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Internal Marks for 5 Courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter marks for course " + (i + 1) + ": ");
            marks[i] = s.nextInt();
        }
    }
}
```

```

// Method to display internal marks
public void displayCIEmarks() {
    System.out.println("Internal Marks: ");
    for (int i = 0; i < 5; i++) {
        System.out.println("Course " + (i + 1) + ": " + marks[i]);
    }
}

package SEE;

import CIE.Internals;
import java.util.Scanner;

public class External extends Internals {
    protected int[] externalMarks = new int[5]; // Array to store external marks for 5 courses
    protected int[] finalMarks = new int[5]; // Array to store final marks (internal + external)

    // Method to input external marks for five courses
    public void inputSEEmarks() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter External Marks for 5 Courses: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Enter marks for course " + (i + 1) + ": ");
            externalMarks[i] = s.nextInt();
        }
    }

    // Method to calculate final marks (internal + external)
    public void calculateFinalMarks() {
        for (int i = 0; i < 5; i++) {
            finalMarks[i] = marks[i] + externalMarks[i]; // Adding internal and external marks
        }
    }

    // Method to display final marks along with student details
    public void displayFinalMarks() {
        displayStudentDetails(); // Display student details
        displayCIEmarks(); // Display internal marks
        System.out.println("External Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + externalMarks[i]);
        }
        System.out.println("Final Marks: ");
        for (int i = 0; i < 5; i++) {
            System.out.println("Course " + (i + 1) + ": " + finalMarks[i]);
        }
    }
}

```

```
import SEE.External;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of students: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume the newline character

        // Array to store External objects
        External[] students = new External[n];

        // Input details for each student
        for (int i = 0; i < n; i++) {
            students[i] = new External();
            System.out.println("Enter details for student " + (i + 1));
            students[i].inputStudentDetails();
            students[i].inputCIEmarks();
            students[i].inputSEEmarks();
        }

        // Calculate and display final marks for each student
        for (int i = 0; i < n; i++) {
            students[i].calculateFinalMarks();
            students[i].displayFinalMarks();
        }

        sc.close();
    }
}
```

LAB - 7

create a package CIE which has 2 classes Student & Internals. The class Person has members like usn, name, sem. The class Internals has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in 5 courses that of the current sem of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in 5 courses of the current semester of the student. Import the 2 packages in a file that displays the final marks of n students in all 5 courses.

import java.util.*;

package CIE;

```
public class Student {  
    public String USN;  
    public String Name;  
    public int Sem;  
    public int Marks = new int[5];  
    Scanner input = new Scanner(System.in);  
    Student (String USN, String Name, int Sem)  
    {  
        this.USN = USN;  
        this.Name = Name;  
        this.Sem = Sem;  
    }  
}
```

```
package CIE; extends Student
public class Internals { Scanner input = new Scanner(System.in);
    public int[] Marks = new int[5];
    public void setMarks () {
        for (int i=0; i<5; i++) {
            System.out.print("Enter the marks of subject " + i + ": ");
            Marks[i] = input.nextInt();
        }
    }
}
```

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

public class External extends Internals {
    protected int[] externalMarks = new int[5];
    protected int[] finalMarks = new int[5];
    Scanner input = new Scanner(System.in);
    public void setMarks() {
        System.out.println("Enter marks of 5 subjects : ");
        for (int i=0; i<5; i++) {
            externalMarks[i] = input.nextInt();
        }
    }
    public void calcFinalMarks() {
        finalMarks
        for (int i=0; i<5; i++) {
            finalMarks[i] = Marks[i] + externalMarks[i];
        }
        System.out.println("Final Marks : ");
    }
}
```

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

```
public class Main {  
    public static void main (String [] args) {  
        int n = 3;  
        for (int i=0; i<n; i++) {  
            External e = new External();  
            students[i] = e;  
            System.out.println ("Name: ");  
            String Name = s.nextLine();  
            Scanner s = new Scanner (System.in);  
            System.out.println ("USN: ");  
            String USN = s.nextLine();  
            System.out.println ("Sem: ");  
            int Marks = s.nextInt();  
            students[i].Student (USN, Name,  
                Sem);  
            students[i].setMarks ();  
            students[i].setMarks (SEE);  
            students[i].calcFinalMarks ();  
        }  
    }  
}
```

OUTPUT:

Enter the number of students: 2

Enter details of student 1

Enter USM:

IBM20CS001

Enter Name :

Alice

Enter Sem :

5

Enter Internal Marks for 5 course :

Enter marks for course 1 :

20

Enter marks for course 2 :

22

Enter marks for course 3 :

18

Enter marks for course 4 :

25

Enter marks for course 5 :

24

Enter External Marks for 5 course :

Enter marks for course 1 :

55

Enter marks for course 2 :

60

Enter marks for course 3 :

50

Enter marks for course 4 :

70

Enter marks for course 5 :

65

Student Details :

USN : IBM20 CS001

Name : Alice

Semester : 5

Internal Marks :

Coue. 1 : 20

Course 2 : 22

Course 3 : 18

Course 4 : 25

Course 5 : 24

External Marks:

Course 1 : 55

Course 2 : 60

Course 3 : 50

Course 4 : 70

Course 5 : 65

Final Marks:

Course 1 : 75

Course 2 : 82

Course 3 : 68

Course 4 : 95

Course 5 : 89

~~classmate~~

Lab8:

Program1:

```
import java.util.*;
interface Polygon{
    void getPerimeter();
    public abstract void getArea();
}
class Shape implements Polygon{
    int n;
    Shape(int n){
        this.n=n;
    }
    public void getPerimeter(){
        int P=0;
        Scanner input = new Scanner(System.in);
        for (int i=0; i<n; i++){
            System.out.println("Enter the length of the side:");
            int l=input.nextInt();
            P+=l;
        }
        System.out.println("The Perimeter is:"+P);
    }
    public void getArea(){
        System.out.println("Area is implemented");
    }
}
public class Main2{
    public static void main(String args[]){
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the no. of sides:");
        int n=input.nextInt();
        Polygon S= new Shape(n);
        S.getPerimeter();
        S.getArea();
    }
}
```

Output:

Enter the no. of sides:

3

Enter the length of the side:

2

Enter the length of the side:

2

Enter the length of the side:

3

The Perimeter is:7

Area is implemented

LAB - 8

Page No. _____
Date _____

Program 1:

Implementation of method 1

Program 2:

Dog barks

Dog eats bones

Program 3:

Sedan is driving starting

Sedan driving

Program 4:

Printing Document

Showing Document Preview

Program 5:

We have created an interface named Polygon.

It includes a default method getPerimeter()
of an abstract method getArea().

We can calculate the perimeter of all polygons
in the same manner so we implemented
the body of getPerimeter() in Polygon.

Now all polygons that implement Polygon
can use getPerimeter() to calculate perimeter.

Hence the rule for calculating the area
is different for different polygons. Hence
getArea() is included without implementation.

Any class that implements Polygon must
provide an implementation of getArea()

SOURCE CODE:

```
import java.util.*;  
  
interface Polygon {  
    void getPerimeter();  
    public abstract void getArea();  
}  
  
class Shape implements Polygon {  
    System.out.println("Enter  
Scanner  
Shape (int n) {  
    this.n = n;  
}  
    public void  
    shape getPerimeter () {  
        Scanner input = new Scanner();  
        for (int i=0; i<n; i++) {  
            int P=0;  
            System.out.println ("Enter the  
length of side  
l: " + input.nextInt());  
            P += l;  
        }  
        System.out.println ("The Perimeter  
is :" + P);  
    }  
    public void getArea () {  
        System.out.println ("Area imple")
```

66
Page No. _____ Date. _____

```
public class Main {
    public static void main (String args[])
    {
        Scanner s = new Scanner();
        System.out.println ("Enter the no. of sides:");
        int n = s.nextInt();
        Polygon Polygon S = new Shape();
        S.getPerimeter();
        S.getArea();
    }
}
```

OUTPUT :

Enter the no. of sides:

3

Enter the length of the side:

2

Enter the length of the side:

2

Enter the length of the side:

3

The Perimeter is : 7

~~Area is implemented~~

side:");
~~Q1~~

imper

dm ("Area implemented"); }

Lab9:

Program1:

```
class Father {  
    int age;  
  
    Father(int age) throws WrongAgeException {  
        if (age <= 0) {  
            throw new WrongAgeException("Father's age can't be less than 0");  
        }  
        this.age = age;  
    }  
}  
  
class Son extends Father {  
    int age;  
  
    Son(int fatherAge, int age) throws WrongAgeException {  
        super(fatherAge);  
        if (age > fatherAge) {  
            throw new WrongAgeException("Son can't be older than father");  
        }  
        this.age = age;  
    }  
}  
  
class WrongAgeException extends Exception {  
    WrongAgeException(String message) {  
        super(message);  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        try {  
            Father f1 = new Father(50);  
            Son s1 = new Son(f1.age, 10);  
  
            Father f2 = new Father(0);  
            Son s2 = new Son(f2.age, 10);  
        } catch (WrongAgeException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

Output:

Father's age can't be less than 0
Son can't be older than father

26/12/24.

66
Page No.
Date: / /

LAB - 9

Write a program that demonstrates handling of exception in inheritance tree. Create a base class called 'Father' of a derived class called 'Son' which extends the base class. In father class, implement a constructor which takes age of throws wrongAge() when the input age < 0. In son class, implement a constructor that uses both father & son's age & throws an exception if son's age is \geq father's age.

SOURCE CODE:

```
class WrongAgeException extends Exception {
```

```
    WrongAgeException (String message) {  
        super (message);  
    }
```

```
class Father {
```

```
    int age;
```

```
    Father (int age) throws wrongAgeException {
```

```
        if (age < 0) {
```

```
            throw new wrongAgeException (  
                "Father's age can't be less than 0");
```

```
        } this.age = age;
```

```
}
```

Date: _____ / _____ / _____

```
class Son extends Father {  
    int age; }  
public Son (int fatherage, int age) {  
    if (age > 0 && age < fatherage) {  
        this.age = age;  
        super (fatherage);  
    } }  
} } use { new  
throw WrongAgeException ("Son can't  
be older than father");
```

```
class Main {  
    public static void main (String args[]) {  
        try {  
            Father f1 = new Father (50);  
            Son s1 = new Son (f1.age, 10);  
            Father f2 = new Father (0);  
            Son s2 = new Son (f1.age, 70);  
        } catch {  
    } }
```

OUTPUT:

~~Father's age can't be less than 0
Son can't be older than father.~~

Q4

Lab10:

Program1:

```
class CollegeThread extends Thread{
    private String message;
    private int interval;
    public CollegeThread(String message,int interval){
        this.message=message;
        this.interval=interval;
    }
    @ Override
    public void run(){
        try{
            while(true){
                System.out.println(message);
                Thread.sleep(interval);
            }
        }
        catch(InterruptedException e){
            System.out.println("Thread Interrupted:" + message);
        }
    }
}
class Main{
    public static void main(String args[]){
        CollegeThread c1=new CollegeThread("BMSCE",10000);
        CollegeThread c2=new CollegeThread("CSE",2000);
        c1.start();
        c2.start();
    }
}
```

Output:

BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
BMSCE

CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE
CSE
CSE
CSE
BMSCE
CSE
CSE
CSE

Program2:

```
import java.util.Scanner;
class Main{
    public static void main(String args[]){
        Scanner s = new Scanner(System.in);
        try{
            System.out.println("Enter num1:");
            int num1=s.nextInt();
            System.out.println("Enter num2:");
            int num2=s.nextInt();
            if (num2==0){
                throw new ArithmeticException("cannot divide by zero");
            }
            System.out.println("Result:"+ (num1/num2));
        }
        catch(ArithmeticException e){
            System.out.println("Error:" + e.getMessage());
        }
        catch(Exception e){
            System.out.println("Error: Enter A VALID INTEGER");
        }
        finally{
            s.close();
        }
    }
}
```

Output:

Enter num1:
5

Enter num2:

6

Result:0

3/12/24

66
Page No. _____
Date: _____ 99

WEEK - 10

- 1) Write a program which creates 2 threads, one thread displaying "BMS College of Engineering" once every 10 s & another "CSE" once every 2 seconds.

SOURCE CODE:

```
import java.util.*;  
  
class CollegeThread extends Thread {  
    private String message;  
    private int interval;  
  
    public CollegeThread (String message, int interval) {  
        this.message = message;  
        this.interval = interval;  
    }  
  
    @Override  
    public void run () {  
        try {  
            while (true) {  
                System.out.println (message);  
                Thread.sleep (interval);  
            }  
        } catch (InterruptedException e) {  
            System.out.println ("Thread Interrupted : " +  
                message);  
        }  
    }  
}
```

Page No. _____
Date. _____

```
public class ThreadMessage
{
    public static void main (String [] args)
    {
        CollegeThread c1 = new CollegeThread
            ("BMSCE", 10000);
        CollegeThread c2 = new CollegeThread ("CSE",
            2000);
        c1.start();
        c2.start();
    }
}
```

OUTPUT :

BMSCE

CSE

CSE

CSE

CSE

CSE

~~BMSCE~~

CSE.

interrupted : "-"

2) Write a program that creates a user interface to perform integer division. The user enters 2 nos. in the text fields. Num1 & Num2. The division of Num1 & Num2 is displayed in the result field when the divide button is clicked. If Num1 or Num2 weren't an integer the program would throw a NumberFormatException. If Num2 were zero, the program would throw an ArithmeticException. Display exception in message box.

SOURCE CODE:

```
import java.util.Scanner;  
  
public class Division {  
    public static void main (String args[]) {  
        Scanner s = new Scanner (System.in);  
  
        try {  
            System.out.print ("Enter num1:");  
            int num1 = s.nextInt ();  
  
            System.out.print ("Enter num2:");  
            int num2 = s.nextInt ();  
  
            if (num2 == 0) {  
                throw new ArithmeticException  
                ("cannot divide by zero");  
            }  
        }  
    }  
}
```

```
int result = num1 / num2;
System.out.println ("Result" + result);
}

catch (ArithmeticException e) {
    System.out.println ("Error" + e.getMessage());
}

catch (Exception e) {
    System.out.println ("Error: Please
enter valid integer");
}

finally {
    s.close();
}
}
```

OUTPUT:

```
Enter num1: 2
Enter num2: 2
Result: 1
Enter num1: 3
Enter num2: 0
Cannot divide by zero
```

Q13

