



**RGM College of Engineering and Technology**  
(Autonomous)  
**Department of Computer Science & Engineering**

# Java Programming

---

*II - B. Tech II- Semester (R-19)*

**Java Programming Lab Manual for the Academic Year  
2020-2021**





### **VISION OF THE DEPARTMENT**

- To empower students with cutting edge technologies in computer science and engineering
- To train the students as entrepreneurs in computer science and engineering to address the needs of the society
- To develop smart applications to disseminate information to rural people

### **MISSION OF THE DEPARTMENT**

- To become the best computer science and engineering department in the region offering undergraduate, post graduate and research programs in collaboration with industry
- To incubate, apply and spread innovative ideas by collaborating with relevant industries and R & D labs through focused research groups.
- To provide exposure to the students in the latest tools and technologies to develop smart applications for the society

# Java Programming

---

Rajeev Gandhi Memorial College Of Engineering College



Nandyal, Kurnool District – 518 501 (A. P.)

---

## **Department of Computer Science and Engineering**

Lab Manual for the Academic Year 2020-21

(In accordance with RGM CET-R19 syllabus)

SUBJECT : Java Programming Lab

SUBJECT CODE : A0598154

SEMESTER : II Year II Semester

STREAM : Computer Science and Engineering

INSTRUCTOR : M. Sravan Kumar Reddy

PROGRAMMER'S : M. Kiran Kumar

# Java Programming

---

## **JAVA PROGRAMMING LAB**

### **OBJECTIVES:**

- ❖ To make the student operating systems.
- ❖ Learn object oriented way of solving problems.
- ❖ To teach the student to write programs in Java to solve the problems

### **OUTCOMES:**

After Completion of the Lab Course student should be able:

- ❖ Student can able to write programs using classes and objects.
- ❖ Student can able to develop the polymorphic behaviour of objects.
- ❖ Students can able to design software using object oriented approach.
- ❖ Able to implement the programs handling built in exceptions and creating custom Exceptions.
- ❖ Able to develop the Multithread programming.

### **Exercise Programs**

#### **(2019 – 2023) II- B.Tech II- Sem (R19)**

1. Write a Java Program to find the reverse of a given number. And also check whether it is palindrome or not.
2. Write a Java Program to print Fibonacci sequence (rule: The first two values in the sequence are 0 and 1. Every subsequent value is the sum of the two values preceding it.)
3. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer.
4. Write a Java program to multiply two given matrices.
5. Write a Java program to find both the largest and smallest number in a list of integers.
6. Write a Java program to find the volume of a box by creating objects.
7. Write a Java program to implement all arithmetic operations with static/class methods for each operation. User must provide values from Keyboard.
8. Write a Java program to implement parameter passing techniques:  
a) call-by-value                      b) call-by-reference
9. Write a Java program to implement the following:  
a) Overloading methods                      b) overloading constructors

## Java Programming

---

10. Write a Java program to implement multi-level inheritance and also demonstrate super keyword.
11. Write a Java program to demonstrate method overriding by implementing dynamic method dispatch?
12. Write a Java program to create an abstract class called Shape which inherits Rectangle and triangle to calculate area of each shape by implementing abstract method of Shape class by implementing hierarchical inheritance.
13. Write a Java program to implement multiple-inheritance?
14. Write a Java program to import user defined packages to display results for any mathematical operations like addition, subtractions, multiplications and division (class methods) from one package and also producing results square, cube and square-root of a given number (instant methods) from another package.
15. Write a Java program to sort a list of names in ascending order.
16. Write a Java program that checks whether a given string is a palindrome or not.  
Ex: MADAM is a palindrome
17. Write a Java program to handle multiple exceptions and also use finally?
18. Write a Java program to handle user-defined exceptions?
19. Write a Java program that reads a file name from the user, then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
20. Write a Java program that displays the count of number of characters, lines and words in a text file.( user provide file using cmd line args)
21. Write a Java program that creates three threads. First thread displays “Good Morning” every one second, the second thread displays “Hello” every two seconds and the third thread displays Welcome” every three seconds.
22. Write a Java program that correctly implements producer consumer problem using the concept of inter-thread communication
23. Write a Java program to store student class objects in Array List and sort the ArrayList?
24. Write a Java program to create HashSet from ArrayList and remove duplicates and display results?
25. Write a java program to traversing elements in descending order in TreeSet.

## Java Programming Lab - 2019-2023 - II Year - II Sem

S.No	Date	Name of the Experiment	Status	Page No	Remarks
1		Write a Java Program to find the reverse of a given number. And also check whether it is palindrome or not. Click Here	✓		
2		Java Program to Display the Fibonacci Series Click Here	✓		
3		Write a Java program that prints out all prime numbers up to given integer. Click Here	✓		
4		Program to find Multiplication of Two matrices Click Here	✓		
5		Find minimum and maximum numbers from given array. Click Here	✓		
6		Write a Java program to find the volume of a box by creating objects. Click Here	✓		
7		Create static methods to perform arithmetic operations and access them in other classes. Click Here	✓		
8		Write a Java program to exchange two variables using call by value Click Here	✓		
9		Write a Java program to exchange two variables using call by reference Click Here	✓		
10		Write a Java program to implement Method overloading Click Here	✓		
11		Write a Java program to implement Constructor overloading Click Here	✓		
12		Write a Java program to implement Multilevel Inheritance Click Here	✓		
13		Write a Java program to achieve concept of Method Overriding Click Here	✓		
14		Write a Java program to import user defined packages to display results for any mathematical operations like addition, subtractions, multiplications and division (class methods) from one package Click Here	✓		

S.No	Date	Name of the Experiment	Status	Page No	Remarks
15		Program to implement abstraction, create a class with one abstract method and implementing them in another class. Click Here	✓		
16		Write a Java program to implement multiple-inheritance. Click Here	✓		
17		Write a Java program to sort a list of names in ascending order. Click Here	✓		
18		Program to check the given String is Palindrome or not Click Here	✓		
19		Write a Java program that handles multiple exceptions. Click Here	✓		
20		Create a user exception and then handles that exception in other program. Click Here	✓		
21		Display information about file Click Here	✓		
22		Count characters, words and lines in a file. Click Here	✓		
23		Write a Java program demonstrating the usage of Threads Click Here	✓		
24		Java program that correctly implements the producer consumer problem Click Here	✓		
25		Iteration on ArrayList Click Here	✓		
26		Iteration on ArrayList Click Here	✓		
27		Write a java program to traversing elements in descending order in TreeSet. Click Here	✓		

S.No: 1

Exp. Name: **Write a Java Program to find the reverse of a given number. And also check whether it is palindrome or not.**

Date:

**Aim:**

Write a Java Program to find the reverse of a given number. And also check whether it is palindrome or not.

**Source Code:**

Palindrome.java

```
import java.util.Scanner;
public class Palindrome
{
    public static void main(String args[])
    {
        int n, m, a = 0, x;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter any number:");
        n = s.nextInt();
        m = n;
        while(n > 0)
        {
            x = n % 10;
            a = a * 10 + x;
            n = n / 10;
        }
        System.out.println("Reverse of a number is "+a);
        if(a == m)
        {
            System.out.println("Given number "+m+" is Palindrome");
        }
        else
        {
            System.out.println("Given number "+m+" is Not Palindrome");
        }
    }
}
```

**Execution Results - All test cases have succeeded!****Test Case - 1****User Output**

Enter any number: 145

Reverse of a number is 541

Given number 145 is Not Palindrome

**Test Case - 2****User Output**

Enter any number: 111

Reverse of a number is 111

Given number 111 is Palindrome



**S.No: 2**Exp. Name: **Java Program to Display the Fibonacci Series****Date:****Aim:**

Write a class `FibonacciSeries` with a **main** method. The method receives one command line argument. Write a program to display **fibonacci series** i.e. `0 1 1 2 3 5 8 13 21.....`

For example:

```
Cmd Args : 80
0 1 1 2 3 5 8 13 21 34 55
```

**Note:** Please don't change the package name.**Source Code:**

q10896/FibonacciSeries.java

```
package q10896;
class FibonacciSeries {
    public static void main(String[] args) {
        int n=Integer.parseInt(args[0]);
        int a=0,b=1;
        System.out.print(a+ " "+b);
        int c=a+b;
        do
        {
            System.out.print(" "+c);

            a=b;
            b=c;
            c=a+b;
        }while(c<n);
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

0 1 1 2 3 5

**Test Case - 2****User Output**

0 1 1 2 3 5 8 13 21 34 55

**Test Case - 3****User Output**

0 1 1 2 3 5 8 13 21 34 55

S.No: 3

Exp. Name: **Write a Java program that prints out all prime numbers up to given integer.**

Date:

**Aim:**

Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that integer. (Use Scanner/ BufferedReader class)

**Source Code:**

PrintPrime.java

```
import java.util.*;
class PrintPrime{
    public static void main(String a[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n=s.nextInt();
        System.out.print("Prime numbers are: ");
        for (int i = 1; i <= n; i++) {
            int counter=0;
            for(int num =i; num>=1; num--) {
                if(i%num==0) {
                    counter = counter + 1;
                }
            }
            if (counter ==2)
            {
                //Appended the Prime number to the String
                System.out.print(i+" ");
            }
        }
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

Enter a number: 50

Prime numbers are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47

**Test Case - 2****User Output**

Enter a number: 25

Prime numbers are: 2 3 5 7 11 13 17 19 23

**Test Case - 3****User Output**

Enter a number: 75

Prime numbers are: 2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73

**S.No: 4** Exp. Name: **Program to find Multiplication of Two matrices****Date:****Aim:**

Write a class `MultiplicationOfMatrix` with a **public** method `multiplication` which returns the multiplication result of its arguments. if the first argument column size is not equal to the row size of the second argument, then the method should return null.

Consider the following example for your understanding

```
Matrix 1:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 4 5
Enter row 3: 7 8
Matrix 2:
Enter number of rows: 2
Enter number of columns: 3
Enter 3 numbers separated by space
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Multiplication of the two given matrices is:
9 12 15
24 33 42
39 54 69
```

```
Matrix 1:
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 3 4
Matrix 2:
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space
Enter row 1: 1 2
Enter row 2: 4 5
Enter row 3: 2 3
Multiplication of matrices is not possible
```

**Note:** Please don't change the package name.

**Source Code:**`q11106/MultiplicationOfMatrix.java`

```
package q11106;
public class MultiplicationOfMatrix{
    public int[][] multiplication(int[][] matrix1, int[][] matrix2) {
        /*Return the result if the matrix1 coloumn size is equal to matrix2 row size and
        print the result.
        * @Return null.
        */
    }
}
```

```
// Write your logic here for matrix multiplication
int r1=matrix1.length;
int r2=matrix2.length;
int c1=matrix1[0].length;
int c2=matrix2[0].length;
if(c1==r2)
{
    int c[][]=new int[r1][c2];
    for (int i = 0; i < r1; i++)
    {
        for (int j = 0; j < c2; j++)
        {
            c[i][j] = 0;
            for (int k = 0; k < c1; k++)
                c[i][j] += matrix1[i][k]* matrix2[k][j];
        }
    }
    return c;
}
else return null;
}
}
```

#### q11106/MultiplicationOfMatrixMain.java

```
package q11106;
import java.util.Scanner;
public class MultiplicationOfMatrixMain {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        MultiplicationOfMatrix multiplier = new MultiplicationOfMatrix();

        System.out.println("Matrix 1:");
        int[][] m1 = readMatrix(s);

        System.out.println("Matrix 2:");
        int[][] m2 = readMatrix(s);

        int[][] multi = multiplier.multiplication(m1, m2);

        if (multi == null) {
            System.out.println("Multiplication of matrices is not possible");
        } else {
            System.out.println("Multiplication of the two given matrices is:");
            for (int i = 0; i < multi.length; i++) {
                int c = multi[i].length;
                for (int j = 0; j < c; j++) {
                    String spacer = j == c - 1 ? "\n" : " ";
                    System.out.print(multi[i][j] + spacer);
                }
            }
        }
    }

    public static int[][] readMatrix(Scanner s) {
```

```

        System.out.print("Enter number of rows: ");
        int r = s.nextInt();
        System.out.print("Enter number of columns: ");
        int c = s.nextInt();
        int[][] m = new int[r][c];
        System.out.println("Enter " + c + " numbers separated by space");
        for (int i = 0; i < r; i++) {
            System.out.print("Enter row " + (i + 1) + ": ");
            for (int j = 0; j < c; j++) {
                m[i][j] = s.nextInt();
            }
        }
        return m;
    }
}

```

### Execution Results - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
Matrix 1: 2
Enter number of rows: 2
Enter number of columns: 3
Enter 3 numbers separated by space 1 2 3
Enter row 1: 1 2 3
Enter row 2: 4 5 6
Matrix 2: 3
Enter number of rows: 3
Enter number of columns: 2
Enter 2 numbers separated by space 1 2
Enter row 1: 1 2
Enter row 2: 3 4
Enter row 3: 5 6
Multiplication of the two given matrices is:
22 28
49 64

Test Case - 2
<b>User Output</b>
Matrix 1: 2
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space 1 2
Enter row 1: 1 2
Enter row 2: 3 4
Matrix 2: 2
Enter number of rows: 2
Enter number of columns: 2
Enter 2 numbers separated by space 5 6
Enter row 1: 5 6
Enter row 2: 7 8

Test Case - 2
Multiplication of the two given matrices is:
19 22
43 50

**S.No: 5**Exp. Name: **Find minimum and maximum numbers from given array.****Date:****Aim:**

Write a Java program to find minimum and maximum numbers in a given array.

**Source Code:**

MinMaxArray.java

```
import java.util.Scanner;
class MinMaxArray{
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n=s.nextInt();
        int arr[]=new int[n];
        System.out.print("Enter array elements: ");
        for(int i=0; i<n; i++){
            arr[i]=s.nextInt();
        }
        int min=arr[0];
        int max=arr[0];
        for(int i=0; i<n; i++){
            if(arr[i]<min)
                min=arr[i];
            if(arr[i]>max)
                max=arr[i];
        }
        System.out.println("Mimimum element in array is: "+min);
        System.out.println("Maximum element in array is: "+max);
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

```
Enter number of elements: 5
Enter array elements: 100 1 135 0 12
Mimimum element in array is: 0
Maximum element in array is: 135
```

**Test Case - 2****User Output**

```
Enter number of elements: 10
Enter array elements: 2 2222 222 22222 22 1 1111 111111 111 11111
Mimimum element in array is: 1
Maximum element in array is: 111111
```

Test Case - 3
User Output
Enter number of elements: 8
Enter array elements: 88 77 66 55 44 33 22 11
Minimum element in array is: 11
Maximum element in array is: 88



**S.No: 6**Exp. Name: **Write a Java program to find the volume of a box by creating objects.****Date:****Aim:**

Write a Java program to find the volume of a box by creating objects.

**Source Code:**

BoxDemo.java

```
import java.util.*;
class BoxDemo
{
    double l,b,h;
    BoxDemo(double l,double b,double h)
    {
        this.l=l;
        this.b=b;
        this.h=h;
    }
    public double vol()
    {
        return (l*b*h);
    }
    public static void main(String args[])
    {
        double l1,b1,h1;
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter the length of box:");
        l1=sc.nextDouble();
        System.out.println("Enter the breadth of box:");
        b1=sc.nextDouble();
        System.out.println("Enter height of box:");
        h1= sc.nextDouble();
        BoxDemo bx=new BoxDemo(l1,b1,h1);
        System.out.println("Volume is "+bx.vol());
    }
}
```

Page No:

ID: CSE016

Rajeev Gandhi Memorial College of Engineering and Technology (Autonomous)

**Execution Results** - All test cases have succeeded!

Test Case - 1
User Output
Enter the length of box: 3
Enter the breadth of box: 6
Enter height of box: 9
Volume is 162.0

**S.No: 7****Exp. Name: *Create static methods to perform arithmetic operations and access them in other classes.*****Date:****Aim:**

Write a java program to create static methods to perform arithmetic operations. Access them in other classes and display results.

**Source Code:**

Arithmetic.java

```
import java.util.*;
class ArithmeticOperations{
    static int add(int a,int b){
        return a+b;
    }
    static int sub(int a,int b){
        return a-b;
    }
    static int mul(int a,int b){
        return a*b;
    }
    static int div(int a,int b){
        if (b==0)
            return 0;
        else
            return a/b;
    }
    static int mod(int a,int b){
        if (b==0)
            return 0;
        else
            return a%b;
    }
}
class Arithmetic{
    public static void main(String a[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter a number: ");
        int n1=s.nextInt();
        System.out.print("Enter another number: ");
        int n2=s.nextInt();
        System.out.println("Addition of "+n1+" and "+n2+" is: "+ArithmeticOperations.add(n1,n2));
        System.out.println("Subtraction of "+n1+" and "+n2+" is: "+ArithmeticOperations.sub(n1,n2));
        System.out.println("Multiplication of "+n1+" and "+n2+" is: "+ArithmeticOperations.mul(n1,n2));
        System.out.println("Division of "+n1+" and "+n2+" is: "+ArithmeticOperations.div(n1,n2));
        System.out.println("Modulo of "+n1+" and "+n2+" is: "+ArithmeticOperations.mod(n1,n2));

    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter a number: 5
Enter another number: 10
Addition of 5 and 10 is: 15
Subtraction of 5 and 10 is: -5
Multiplication of 5 and 10 is: 50
Division of 5 and 10 is: 0
Modulo of 5 and 10 is: 5

Test Case - 2
User Output
Enter a number: 50
Enter another number: 90
Addition of 50 and 90 is: 140
Subtraction of 50 and 90 is: -40
Multiplication of 50 and 90 is: 4500
Division of 50 and 90 is: 0
Modulo of 50 and 90 is: 50

S.No: 8

Exp. Name: **Write a Java program to exchange two variables using call by value**

Date:

**Aim:**

Write a Java program to exchange two variables using call by value

**Source Code:**

SwapDemo.java

```
import java.util.*;
class SwapDemo
{
    public void swap(int i,int j)
    {
        int temp =i;
        j=i;
        temp=i;
    }

    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int a,b;
        System.out.println("Enter value 1:");
        a=sc.nextInt();
        System.out.println("Enter value 2:");
        b=sc.nextInt();
        SwapDemo sd=new SwapDemo();
        System.out.println("Before swapping value of a is "+a+ " value of b is "+b);
        sd.swap(a,b);
        System.out.println("After swapping value of a is "+a+ " value of b is "+b);
    }
}
```

**Execution Results** - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
Enter value 1: 10
Enter value 2: 20
Before swapping value of a is 10 value of b is 20
After swapping value of a is 10 value of b is 20

**S.No: 9**Exp. Name: **Write a Java program to exchange two variables using call by reference****Date:****Aim:**

Write a Java program to exchange two variables using call by reference

**Source Code:**

SwapDemo1.java

```
import java.util.Scanner;
class SwapDemo1
{
    int x,y;
    SwapDemo1(int i, int j){
        this.x=i;
        this.y=j;
    }
    void swap(SwapDemo1 o){
        int temp=o.x;
        this.x=o.y;
        this.y=temp;
    }
    public static void main(String a[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter value 1:");
        int i1=sc.nextInt();
        System.out.println("Enter value 2:");
        int i2=sc.nextInt();
        SwapDemo1 s=new SwapDemo1(i1,i2);
        System.out.println("Before swapping value of a is "+s.x+" value of b is "+s.y);
        s.swap(s);
        System.out.println("After swapping value of a is "+s.x+" value of b is "+s.y);
    }
}
```

**Execution Results** - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
Enter value 1: 10
Enter value 2: 20
Before swapping value of a is 10 value of b is 20
After swapping value of a is 20 value of b is 10

S.No: 10 Exp. Name: **Write a Java program to implement Method overloading**

Date:

**Aim:**

Write a Java program with a class name `Addition` with the methods `add(int, int)`, `add(int, float)`, `add(float, float)` and `add(float, double, double)` to add values of different argument types.

Write the **main(String[])** method within the class and assume that it will always receive a total of **6** command line arguments at least, such that the first **2** are **int**, next **2** are **float** and the last **2** are of type **double**.

If the **main()** is provided with arguments : **1, 2, 1.5f, 2.5f, 1.0, 2.0** then the program should print the output as:

```
Sum of 1 and 2 : 3
Sum of 1.5 and 2.5 : 4.0
Sum of 2 and 2.5 : 4.5
Sum of 1.5, 1.0 and 2.0 : 4.5
```

**Note:** Please don't change the package name.

**Source Code:**

q11266/Addition.java

```
package q11266;
class Addition{
    void add(int x, int y){
        System.out.println("Sum of "+x+" and "+y+" : "+(x+y));
    }
    void add(int x, float y){
        System.out.println("Sum of "+x+" and "+y+" : "+(x+y));
    }
    void add(float x, float y){
        System.out.println("Sum of "+x+" and "+y+" : "+(x+y));
    }
    void add(float x, double y, double z){
        System.out.println("Sum of "+x+", "+y+" and "+z+" : "+(x+y+z));
    }
    public static void main(String a[]){
        int i1=Integer.parseInt(a[0]);
        int i2=Integer.parseInt(a[1]);
        float f1=Float.parseFloat(a[2]);
        float f2=Float.parseFloat(a[3]);
        double d1=Double.parseDouble(a[4]);
        double d2=Double.parseDouble(a[5]);
        Addition a1=new Addition();
        a1.add(i1,i2);
        a1.add(f1,f2);
        a1.add(i2,f2);
        a1.add(f1,d1,d2);
    }
}
```

**Execution Results** - All test cases have succeeded!

Test Case - 1
User Output
Sum of 2 and 1 : 3
Sum of 5.0 and 3.6 : 8.6
Sum of 1 and 3.6 : 4.6
Sum of 5.0, 9.2 and 5.26 : 19.46



S.No: 11

Exp. Name: **Write a Java program to implement Constructor overloading**

Date:

**Aim:**

Write a class `Box` which contains the data members **width**, **height** and **depth** all of type **double**.

Write the implementation for the below **3 overloaded constructors** in the class `Box` :

- **Box()** - default constructor which initializes all the members with **-1**
- **Box(length)** - parameterized constructor with one argument and initialize all the members with the value in **length**  
the members with the corresponding arguments
- **Box(width, height, depth)** - parameterized constructor with three arguments and initialize

Write a method `public double volume()` in the class `Box` to find out the **volume** of the given box.

Write the **main** method within the `Box` class and assume that it will receive either **zero** arguments, or **one** argument or **three** arguments.

For example, if the **main()** method is passed **zero** arguments then the program should print the output as:

```
Volume of Box() is : -1.0
```

Similarly, if the **main()** method is passed **one** argument : **2.34**, then the program should print the output as:

```
Volume of Box(2.34) is : 12.812903999999998
```

then the program should print the output as: Likewise, if the **main()** method is passed **three** arguments : **2.34**, **3.45**, **1.59**, then the program should print the output as:

```
Volume of Box(2.34, 3.45, 1.59) is : 12.836070000000001
```

**Note:** Please don't change the package name.

**Source Code:**

q11267/Box.java

```
package q11267;
public class Box{
    private double width,height,depth;
    public Box(){
        width=height=depth=-1;
    }
    public Box(double length){
        width=height=depth=length;
    }
    public Box(double width,double height, double depth){
        this.width= width;
        this.height=height;
        this.depth=depth;
    }
    public double volume(){
        return width*height*depth;
    }
    public static void main(String args[])throws Exception{

        int n=args.length;
```

```
        if(n==0)
        {

            System.out.println("Volume of Box() is : "+new Box().volume());
        }
        if(n==1)
        {
            double l=Double.parseDouble(args[0]);
            System.out.println("Volume of Box("+l+") is : "+new Box(l).volume());
        }
        if(n==3)
        {
            double l=Double.parseDouble(args[0]);
            double b=Double.parseDouble(args[1]);
            double h=Double.parseDouble(args[2]);
            System.out.println("Volume of Box("+l+", "+b+", "+h+") is : "+new Box(l,b,h).volume());
        }
    }
}
```

### Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Volume of Box() is : -1.0

Test Case - 2
User Output
Volume of Box(3.0) is : 27.0

S.No: 12

Exp. Name: **Write a Java program to implement Multilevel Inheritance**

Date:

**Aim:**

Write a Java program to illustrate the **multilevel inheritance** concept.

Create a class **Student**

- contains the data members **id** of **int** data type and **name** of **string** type
- write a method **setData()** to initialize the data members
- write a method **displayData()** which will display the given **id** and **name**

Create a class **Marks** which is derived from the class **Student**

- contains the data members **javaMarks**, **cMarks** and **cppMarks** of **float** data type
- write a method **setMarks()** to initialize the data members
- write a method **displayMarks()** which will display the given data

Create another class **Result** which is derived from the class **Marks**

- contains the data members **total** and **avg** of **float** data type
- write a method **compute()** to find total and average of the given marks
- write a method **showResult()** which will display the total and avg marks

Write a class **MultilevelInheritanceDemo** with the **main()** method which will receive five arguments as **id**, **name**, **javaMarks**, **cMarks** and **cppMarks**.

Create object only to the class **Result** to access the methods.

If the input is given as command line arguments to the **main()** as **"99", "Lakshmi", "55.5", "78.5", "72"** then the program should print the output as:

```
Id : 99
Name : Lakshmi
Java marks : 55.5
C marks : 78.5
Cpp marks : 72.0
Total : 206.0
Avg : 68.666664
```

**Note:** Please don't change the package name.

**Source Code:**

q11264/MultilevelInheritanceDemo.java

```
package q11264;
class Student{
    int id;
    String name;
    void setData(int i,String n){
        id=i;name=n;
    }
    void displayData(){
        System.out.println("Id : "+id);
        System.out.println("Name : "+name);
    }
}
class Marks extends Student{
    //int id;
    float javaMarks,cMarks,cppMarks;
```

```

void setMarks(float javaMarks, float cMarks, float cppMarks){
    //this.id=id;
    this.javaMarks=javaMarks;
    this.cMarks=cMarks;
    this.cppMarks=cppMarks;
}
void displayMarks(){
    //System.out.println("Id : "+id);
    displayData();
    System.out.println("Java marks : "+javaMarks);
    System.out.println("C marks : "+cMarks);
    System.out.println("Cpp marks : "+cppMarks);
}
}
class Result extends Marks{
    float total,avg;
    void compute()
    {
        total=javaMarks+cMarks+cppMarks;
        avg=total/3;
    }
    void showResult()
    {
        displayMarks();
        System.out.println("Total : "+total);
        System.out.println("Avg : "+avg);
    }
}
class MultilevelInheritanceDemo{
    public static void main(String a[]){
        Result r=new Result();
        int i=Integer.parseInt(a[0]);
        String s=a[1];
        float s1=Float.parseFloat(a[2]);
        float s2=Float.parseFloat(a[3]);
        float s3=Float.parseFloat(a[4]);
        r.setData(i,s);
        r.setMarks(s1,s2,s3);
        r.compute();
        r.showResult();
    }
}

```

### Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Id : 99
Name : Geetha
Java marks : 56.0
C marks : 75.5
Cpp marks : 66.6
Total : 198.1

**Test Case - 1**

Avg : 66.03333

**Test Case - 2****User Output**

Id : 199

Name : Lakshmi

Java marks : 55.5

C marks : 78.5

Cpp marks : 78.0

Total : 212.0

Avg : 70.666664

**S.No: 13** Exp. Name: **Write a Java program to achieve concept of Method Overriding****Date:****Aim:**

Assume there is a class called `Bank` with method `calculateInterest(float principal, int time)`.

Create sub-classes of `Bank` with names `SBI`, `ICICI` and `AXIS` and override the `calculateInterest(float principal, int time)` method.

Create a constant of type `float` called `INTEREST_RATE` in classes `SBI`, `ICICI` and `AXIS` with values `10.8`, `11.6` and `12.3` respectively.

Use the formula  $(\text{principal} * \text{INTEREST\_RATE} * \text{time}) / 100$  to calculate the **interest** for given **principal** and **time** and return the value as `float` in the **overridden method**.

For example, if the **two** arguments passed to the **main** method are 1000 and 5, (principal and time) below is the expected output:

```
SBI rate of interest = 540.0
ICICI rate of interest = 580.0
AXIS rate of interest = 615.0
```

**Note:** Please don't change the package name.

**Source Code:**

q11271/TestOverriding.java

```
package q11271;
class Bank {
    float calculateInterest(float principal, int time) {
        return 0;
    }
}
class SBI extends Bank {
    private static final float INTEREST_RATE = 10.8f;
    float calculateInterest(float principal, int time) {
        return ((principal * INTEREST_RATE * time) / 100);
    }
}
class ICICI extends Bank {
    private static final float INTEREST_RATE = 11.6f;
    float calculateInterest(float principal, int time) {
        return ((principal * INTEREST_RATE * time) / 100);
    }
}
class AXIS extends Bank {
    private static final float INTEREST_RATE = 12.3f;
    float calculateInterest(float principal, int time) {
        return ((principal * INTEREST_RATE * time) / 100);
    }
}

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

```

    Bank sbiBank = new SBI();
    Bank iciciBank = new ICICI();
    Bank axisBank = new AXIS();
    float principal = Float.parseFloat(args[0]);
    int time = Integer.parseInt(args[1]);
    System.out.println("SBI rate of interest = " +sbiBank.calculateInterest(principa
1,time) );
    System.out.println("ICICI rate of interest = " +iciciBank.calculateInterest(princ
ipal,time) );
    System.out.println("AXIS rate of interest = " +axisBank.calculateInterest(princip
al,time) );
    }
}

```

### Execution Results - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
SBI rate of interest = 1804.9608
ICICI rate of interest = 1938.6616
AXIS rate of interest = 2055.65

Test Case - 2
<b>User Output</b>
SBI rate of interest = 540.0
ICICI rate of interest = 580.0
AXIS rate of interest = 615.0

Test Case - 3
<b>User Output</b>
SBI rate of interest = 2149.3447
ICICI rate of interest = 2308.5554
AXIS rate of interest = 2447.8647

Test Case - 4
<b>User Output</b>
SBI rate of interest = 648.0
ICICI rate of interest = 696.0
AXIS rate of interest = 738.0

Test Case - 5
<b>User Output</b>
SBI rate of interest = 75600.0
ICICI rate of interest = 81200.0
AXIS rate of interest = 86100.0

<b>S.No: 14</b>	Exp. Name: <b>Write a Java program to import user defined packages to display results for any mathematical operations like addition, subtractions, multiplications and division (class methods) from one package</b>	<b>Date:</b>
-----------------	--	--------------

**Aim:**

Write a Java program to import user defined packages to display results for any mathematical operations like addition, subtractions, multiplications and division (class methods) from one package and also producing results square, cube and square-root of a given number (instant methods) from another package.

**Source Code:**

p1/Arithmetic.java

```
package p1;
public class Arithmetic
{
    public static int add(int x,int y)
    {
        return x+y;
    }
    public static int sub(int x,int y)
    {
        return x-y;
    }
    public static int multiplication(int x,int y)
    {
        return x*y;
    }
    public static int div(int x,int y)
    {
        return y!=0?x/y:1;
    }
}
```

p2/Calculations.java

```
package p2;
public class Calculations
{
    public double square(int x)
    {
        return x*x;
    }
    public double cube(int x)
    {
        return x*x*x;
    }
    public double squareroot(int x)
    {
        return Math.sqrt(x);
    }
}
```



## Operations.java

```
import p1.Arithmetic;
import p2.Calculations;
import java.util.*;
class Operations
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        /*System.out.println("1. addition :");
        System.out.println("2. subtraction:");
        System.out.println("3. multiplication:");
        System.out.println("4. division:");
        System.out.println("5. square:");
        System.out.println("6. cube:");
        System.out.println("7. square-root:");*/
        System.out.println(" 1. addition \n 2. subtraction \n 3. multiplication \n 4. divisio
n"+" \n 5. square \n 6. cube \n 7. square-root \n Enter your Choice:");
        int ch = s.nextInt();
        int a=0,b=0;
        if(ch>0 && ch<=4)
        {
            System.out.println("Enter Two Integer Numbers: ");
            a = s.nextInt();
            b = s.nextInt();
        }
        else if (ch>4 && ch<=7)
        {
            System.out.println("Enter an Integer Number: ");
            a = s.nextInt();
        }
        else{
            System.out.println("Invalid choice");
        }

        switch(ch)
        {
            case 1:
                System.out.println("Addition of " + a + " and "+ b + " is : "+ Arithmetic.add(a,
b));
                break;

            case 2:
                System.out.println("Subtraction of " + a + " and "+ b + " is : "+ Arithmetic.sub
(a,b));
                break;

            case 3:
                System.out.println("Multiplication of " + a + " and "+ b + " is : "+ Arithmetic.m
ultiplication(a,b));
                break;

            case 4:
                System.out.println("Division of " + a + " and "+ b + " is : "+ Arithmetic.div(a,
```

```

b));
    break;

    case 5:
        System.out.println("Square of " + a + " is : " + new Calculations().square(a));
        break;

    case 6:
        System.out.println("Cube of " + a + " is : " + new Calculations().cube(a));
        break;

    case 7:
        System.out.println("Square-root of " + a + " is : " + new Calculations().squareroot(a));
        break;

    }
}

}

/* System.out.println(" 1.addition :");
   System.out.println("2.subtraction:");
   System.out.println("3.multiplication:");
   System.out.println("4.division:");
   System.out.println("5.square:");
   System.out.println("6.cube:");
   System.out.println("7.square-root:");*/

```

### Execution Results - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
1. addition 1
2. subtraction 1
3. multiplication 1
4. division 1
5. square 1
6. cube 1
7. square-root 1
Enter your Choice: 1
Enter Two Integer Numbers: 55 63
Addition of 55 and 63 is : 118

Test Case - 2
<b>User Output</b>
1. addition 2
2. subtraction 2
3. multiplication 2
4. division 2

**Test Case - 2**

5. square 2

6. cube 2

7. square-root 2

Enter your Choice: 2

Enter Two Integer Numbers: 55 96

Subtraction of 55 and 96 is : -41

**Test Case - 3****User Output**

1. addition 3

2. subtraction 3

3. multiplication 3

4. division 3

5. square 3

6. cube 3

7. square-root 3

Enter your Choice: 3

Enter Two Integer Numbers: 5 6

Multiplication of 5 and 6 is : 30

**Test Case - 4****User Output**

1. addition 4

2. subtraction 4

3. multiplication 4

4. division 4

5. square 4

6. cube 4

7. square-root 4

Enter your Choice: 4

Enter Two Integer Numbers: 5 3

Division of 5 and 3 is : 1

**Test Case - 5****User Output**

1. addition 5

2. subtraction 5

3. multiplication 5

4. division 5

5. square 5

6. cube 5

7. square-root 5

Enter your Choice: 5

Enter an Integer Number: 9

Square of 9 is : 81.0

**Test Case - 6****User Output**

**Test Case - 6**

1. addition 6
2. subtraction 6
3. multiplication 6
4. division 6
5. square 6
6. cube 6
7. square-root 6
Enter your Choice: 6
Enter an Integer Number: 6
Cube of 6 is : 216.0

**Test Case - 7****User Output**

1. addition 7
2. subtraction 7
3. multiplication 7
4. division 7
5. square 7
6. cube 7
7. square-root 7
Enter your Choice: 7
Enter an Integer Number: 9
Square-root of 9 is : 3.0

**Test Case - 8****User Output**

1. addition 3
2. subtraction 3
3. multiplication 3
4. division 3
5. square 3
6. cube 3
7. square-root 3
Enter your Choice: 3
Enter Two Integer Numbers: 66 36
Multiplication of 66 and 36 is : 2376

**Test Case - 9****User Output**

1. addition 9
2. subtraction 9
3. multiplication 9
4. division 9
5. square 9
6. cube 9
7. square-root 9
Enter your Choice: 9
Invalid choice

**S.No: 15****Exp. Name: *Program to implement abstraction, create a class with one abstract method and implementing them in another class.*****Date:****Aim:**

Write a java program to implement abstraction. Create a class called Figure which contains two variables dim1, dim2 and one abstract method area () of double type. Define two classes from Figure class namely Rectangle and Triangle respectively and implement such abstract method and display the results.

**Source Code:**

CalcArea.java

```
import java.util.*;
abstract class Figure{
    double dim1,dim2;
    Figure(double d1, double d2){
        dim1=d1;
        dim2=d2;
    }
    abstract double area();
}
class Rectangle extends Figure{
    Rectangle(double d1,double d2){
        super(d1,d2);
    }
    double area(){
        return dim1*dim2;
    }
}
class Triangle extends Figure{
    Triangle(double d1,double d2){
        super(d1,d2);
    }
    double area(){
        return 0.5*dim1*dim2;
    }
}
class CalcArea{
    public static void main(String a[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter length of rectangle: ");
        double d1=s.nextDouble();
        System.out.print("Enter breadth of rectangle: ");
        double d2=s.nextDouble();
        Figure f1=new Rectangle(d1,d2);
        System.out.println("The area of rectangle with sides "+d1+" and "+d2+" is: "+f1.a
rea());

        System.out.print("Enter base of triangle: ");
        double b=s.nextDouble();
        System.out.print("Enter height of triangle: ");
        double h=s.nextDouble();
        f1=new Triangle(b,h);
        System.out.println("The area of rectangle with sides "+b+" and "+h+" is: "+f1.are
a());
```

```
}  
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter length of rectangle: 100
Enter breadth of rectangle: 20
The area of rectangle with sides 100.0 and 20.0 is: 2000.0 50
Enter base of triangle: 50
Enter height of triangle: 82
The area of rectangle with sides 50.0 and 82.0 is: 2050.0

Test Case - 2
User Output
Enter length of rectangle: 54
Enter breadth of rectangle: 20
The area of rectangle with sides 54.0 and 20.0 is: 1080.0 35
Enter base of triangle: 35
Enter height of triangle: 50
The area of rectangle with sides 35.0 and 50.0 is: 875.0

Test Case - 3
User Output
Enter length of rectangle: 5
Enter breadth of rectangle: 10
The area of rectangle with sides 5.0 and 10.0 is: 50.0 5
Enter base of triangle: 5
Enter height of triangle: 10
The area of rectangle with sides 5.0 and 10.0 is: 25.0

**S.No: 16** Exp. Name: **Write a Java program to implement multiple-inheritance.****Date:****Aim:**

Write a Java program to implement multiple-inheritance.

**Source Code:**

MultipleInheritance.java

```
import java.util.Scanner;
interface I1{
    void setData(int l,int b);
}
interface I2{
    int perimeter();
    int area();
}
class MultipleInheritance implements I1,I2{
    int l,b;
    public void setData(int i,int j){
        l=i;
        b=j;
    }
    public int perimeter(){
        if(l==b)
            return 4*l;
        else
            return 2*(l+b);
    }
    public int area(){
        return l*b;
    }
    public static void main(String args[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter length of rectangle 1: ");
        int l=s.nextInt();
        System.out.print("Enter breadth of rectangle 1: ");
        int b=s.nextInt();
        MultipleInheritance mi1=new MultipleInheritance();
        mi1.setData(l,b);
        System.out.println("Perimeter of rectangle 1 is: "+mi1.perimeter());
        System.out.println("Area of rectangle 1 is: "+mi1.area());

        System.out.print("Enter length of rectangle 2: ");
        int l1=s.nextInt();
        System.out.print("Enter breadth of rectangle 2: ");
        int b1=s.nextInt();
        MultipleInheritance mi2= new MultipleInheritance();
        mi2.setData(l1,b1);
        System.out.println("Perimeter of rectangle 2 is: "+mi2.perimeter());
        System.out.println("Area of rectangle 2 is: "+mi2.area());
    }
}
```

**Execution Results** - All test cases have succeeded!

**Test Case - 1****User Output**

Enter length of rectangle 1: 25  
Enter breadth of rectangle 1: 90  
Perimeter of rectangle 1 is: 230 70  
Area of rectangle 1 is: 2250 70  
Enter length of rectangle 2: 70  
Enter breadth of rectangle 2: 70  
Perimeter of rectangle 2 is: 280  
Area of rectangle 2 is: 4900

**Test Case - 2****User Output**

Enter length of rectangle 1: 360  
Enter breadth of rectangle 1: 20  
Perimeter of rectangle 1 is: 760 15  
Area of rectangle 1 is: 7200 15  
Enter length of rectangle 2: 15  
Enter breadth of rectangle 2: 25  
Perimeter of rectangle 2 is: 80  
Area of rectangle 2 is: 375



S.No: 17	Exp. Name: <i>Write a Java program to sort a list of names in ascending order.</i>	Date:
----------	--	-------

**Aim:**  
Write a Java program to sort a list of names in ascending order.

**Source Code:**

SortNames.java

```
import java.util.*;
class SortNames{
    public static void main(String[] args)
    {
        int n;
        String temp;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter the number of names you want to enter: ");
        n = s.nextInt();
        String names[] = new String[n];
        Scanner s1 = new Scanner(System.in);
        System.out.print("Enter the names: ");
        for(int i = 0; i < n; i++)
        {
            names[i] = s1.nextLine();
        }
        for (int i = 0; i < n; i++)
        {
            for (int j = i + 1; j < n; j++)
            {
                if (names[i].compareTo(names[j])>0)
                {
                    temp = names[i];
                    names[i] = names[j];
                    names[j] = temp;
                }
            }
        }
        System.out.print("Sorted names:");
        for (int i = 0; i <= n - 1; i++)
        {
            System.out.print(names[i] + " ");
        }
        //System.out.print(names[n - 1]);
    }
}
```

**Execution Results - All test cases have succeeded!**

Test Case - 1	
<b>User Output</b>	
Enter the number of names you want to enter: 5	
Enter the names: Roy	
Dora	Dora

Test Case - 1	
Zoya	Zoya
Suzan	Suzan
Harry	Harry
Sorted names:Dora Harry Roy Suzan Zoya	

Test Case - 2	
User Output	
Enter the number of names you want to enter: 10	
Enter the names: Matt	
Lionel	Lionel
Gary	Gary
Daniel	Daniel
Chole	Chole
Andreas	Andreas
Sergio	Sergio
Eden	Eden
zenith	zenith
Tim	Tim
Sorted names:Andreas Chole Daniel Eden Gary Lionel Matt Sergio Tim zenith	

Test Case - 3	
User Output	
Enter the number of names you want to enter: 7	
Enter the names: Jhon	
Jack	Jack
Jhonson	Jhonson
Jarden	Jarden
Jockey	Jockey
Jammy	Jammy
Jacky	Jacky
Sorted names:Jack Jacky Jammy Jarden Jhon Jhonson Jockey	

S.No: 18 Exp. Name: **Program to check the given String is Palindrome or not**

Date:

**Aim:**

Create a class `PalindromeOrNot` with a **main** method. The method receives one command line argument. Check the given argument is palindrome or not.

For example:

Cmd Args : madam  
The given string madam is a palindrome

Cmd Args : Godavari  
The given string Godavari is not a palindrome

**Note:** Please don't change the package name.

**Source Code:**

q11184/PalindromeOrNot.java

```
package q11184;
class PalindromeOrNot {
    public static void main(String ar[]) throws Exception{
        String s=ar[0];
        String temp=s;
        StringBuffer str=new StringBuffer(s);
        temp=str.reverse().toString();
        if(temp.equals(s))
            System.out.println("The given string "+s+" is a palindrome");
        else
            System.out.println("The given string "+s+" is not a palindrome");
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

The given string madam is a palindrome

**Test Case - 2****User Output**

The given string Godavari is not a palindrome

**Test Case - 3****User Output**

The given string malayalam is a palindrome

**Test Case - 4****User Output**

The given string 12345 is not a palindrome

S.No: 19 Exp. Name: **Write a Java program that handles multiple exceptions.**

Date:

**Aim:**

Write a Java program that handles multiple exceptions.

**Source Code:**

CatchMultipleException.java

```
import java.lang.*;
import java.util.*;
import java.io.*;

class CatchMultipleException {
    public static void main(String args[]) {
        int num1, num2;
        int arr[] = new int[10];
        Scanner sc = new Scanner(System.in);
        try {
            /*Take values for numerator and denominator from the console */
            System.out.print("Enter a value for numerator: ");
            num1=sc.nextInt();
            System.out.print("Enter a value for denominator: ");
            num2=sc.nextInt();

            arr[10] = num1/num2 ;
        }

        /* catech ArithmeticException and ArrayIndexOutOfBoundsException here*/
        catch(ArithmeticException ae){
            System.out.println("Exception caught: "+ae.getMessage());
        }
        catch(ArrayIndexOutOfBoundsException ai){
            System.out.println("Exception caught: "+ai.getMessage());
        }
    }
}
```

**Execution Results - All test cases have succeeded!****Test Case - 1****User Output**

```
Enter a value for numerator: 15
Enter a value for denominator: 0
Exception caught: / by zero
```

**Test Case - 2****User Output**

```
Enter a value for numerator: 30
Enter a value for denominator: 2
Exception caught: Index 10 out of bounds for length 10
```

S.No: 20

Exp. Name: **Create a user exception and then handles that exception in other program.**

Date:

**Aim:**

Write a Java program to handle user defined exception. Create a user exception and then handle that exception in other program.

**Source Code:**

InvalidWeight.java

```
import java.util.*;
import java.lang.*;

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

CheckWeight.java

```
/* Create a method validWeight which checks if the weight is more than 100 and
throws InvalidWeight exception otherwise. */
import java.util.Scanner;
public class CheckWeight {
    public static void validWeight(int wt) {
        try{
            if(wt>100)
                throw new InvalidWeight(wt+" is invalid weight");
            else
                System.out.println(wt+" is the valid weight.");
        }
        catch(InvalidWeight iw) {
            System.out.println("Exception caught: "+iw.getMessage());
        }
    }

    public static void main(String ar[]){
        Scanner s=new Scanner(System.in);
        System.out.print("Enter weight: ");
        int wt=s.nextInt();
        validWeight(wt);
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

Enter weight: 101

Exception caught: 101 is invalid weight

**Test Case - 2****User Output**

Enter weight: 99

99 is the valid weight.

**Test Case - 3****User Output**

Enter weight: 1000

Exception caught: 1000 is invalid weight

**S.No: 21**Exp. Name: **Display information about file****Date:****Aim:**

Write a Java program that reads a file name from the user then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

**Source Code:**

FileInfo.java

```
import java.io.*;
class FileInfo
{
    public static void main(String args[])throws Exception
    {
        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter file name(text1.txt or text2.txt): ");
        String fname=br.readLine();
        File f=new File(fname);
        System.out.println("File existence: " +f.exists());
        System.out.println("File writable: " +f.canWrite());
        System.out.println("File readable: " +f.canRead());
        System.out.println("File length(in bytes): " +f.length() + " Bytes");

    }
}
```

text1.txt

We know that we cannot save our environment overnight. But, having an intention to make possible is all that count. Reduce wastage of papers. Try not to ruin plants. They are one of the biggest factors that can keep the earth fertile and save the life. Save trees to save our environment. Encourage your child to plant trees in any empty ground. As I said before, the initiative counts. If you feel the need of saving the environment, you will stop others from doing something which can hurt the well being of the nature.

text2.txt

Save Our Environment by Being Responsible:  
Your every action will count. You should not only hold others responsible, make yourself responsible too.  
Why not start saving our environment being a little less self-concerned. Sometimes give priority to the nature before giving priority to yourself. Save the energy, save plants and be sympathetic to the nature surrounding us.

**Execution Results** - All test cases have succeeded!



**Test Case - 1****User Output**

Enter file name(text1.txt or text2.txt): text.txt

File existance: false

File writable: false

File readable: false

File length(in bytes): 0 Bytes

**Test Case - 2****User Output**

Enter file name(text1.txt or text2.txt): text1.txt

File existance: true

File writable: true

File readable: true

File length(in bytes): 521 Bytes

**S.No: 22**Exp. Name: **Count characters, words and lines in a file.****Date:****Aim:**

Write a Java program that displays the number of characters, lines and words in a text file.

**Source Code:**

CountFile.java

```
import java.io.*;
import java.util.*;
class CountFile
{
    public static void main(String args[])throws Exception
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter file name(text1.txt or text2.txt): ");
        String fname=sc.next();
        int cno=0;
        int lno=0;
        int wno=0;
        BufferedReader br=null;
        try
        {
            br=new BufferedReader(new FileReader(fname));

        }
        catch(FileNotFoundException fe)
        {
            System.out.println("File not found.");
        }
        String str;
        while((str=br.readLine())!=null)
        {
            lno++;
            String words[]=str.split(" ");
            for(String s:words)
            {
                cno=cno+s.length();
            }
            wno=wno+words.length;

        }
        System.out.println("Lines: " +lno);
        System.out.println("Words: " +wno);
        System.out.println("Characters: " +cno);
        br.close();
    }
}
```

text1.txt

**Save Our Environment by Being Responsible:**

Your every action will count. You should not only hold others responsible, make yourself responsible too.

Why not start saving our environment being a little less self-concerned. Sometimes give priority to the nature before giving priority to yourself. Save the energy, save plants and be sympathetic to the nature surrounding us.

Life priorities and necessities are never going to reduce. But among all of them, make some time for ensuring the well being of the environment you live in. To save our environment, no life changing movement is required.

If anything is required, that is will power, honest inclination and some small initiatives. Save our environment by being a responsible citizen. Teach your child and others to save water. Do not waste water.

It is a very precious element of our environment.

text2.txt

So we should try to save our environment by making the small day to day initiatives. The first thing you should do is try to save water, trees and electricity. This will make a big difference. Also try to spread good words and educate children about it. Love the nature to save the earth for our own future.

**Execution Results - All test cases have succeeded!**

Test Case - 1
<b>User Output</b>
Enter file name(text1.txt or text2.txt): text1.txt
Lines: 12
Words: 140
Characters: 715

Test Case - 2
<b>User Output</b>
Enter file name(text1.txt or text2.txt): Text1.txt
File not found.

S.No: 23

Exp. Name: **Write a Java program demonstrating the usage of Threads**

Date:

**Aim:**

Write a Java program that uses three threads to perform the below actions:

1. First thread should print "Good morning" for every 1 second for 2 times
2. Second thread should print "Hello" for every 1 seconds for 2 times
3. Third thread should print "Welcome" for every 3 seconds for 1 times

Write appropriate **constructor** in the `Printer` class which implements `Runnable` interface to take three arguments : **message**, **delay** and **count** of types **String**, **int** and **int** respectively.

Write code in the `Printer.run()` method to print the **message** with appropriate **delay** and for number of times mentioned in **count**.

Write a class called `ThreadDemo` with the `main()` method which instantiates and executes three instances of the above mentioned `Printer` class as threads to produce the desired output.

**[Note:** If you want to sleep for 2 seconds you should call `Thread.sleep(2000);` as the `Thread.sleep(...)` method takes milliseconds as argument.]

**Note:** Please don't change the package name.

**Source Code:**

q11349/ThreadDemo.java

```
package q11349;
public class ThreadDemo {
    public static void main(String[] args) throws Exception {
        Thread t1 = new Thread(new Printer("Good morning", 1, 2));
        Thread t2 = new Thread(new Printer("Hello", 1, 2));
        Thread t3 = new Thread(new Printer("Welcome", 3, 1));
        t1.start();
        t2.start();
        t3.start();
        t1.join();
        t2.join();
        t3.join();
        System.out.println("All the three threads t1, t2 and t3 have completed execution.");
    }
}
class Printer implements Runnable {
    String message;
    int delay;
    int count;
    Printer(String message, int count, int delay){
        this.message=message;
        this.delay=delay;
        this.count=count;
    }

    public void run(){
        int time=count*1000;

        try{
            Thread.sleep(time);
```

```
        for(int i=0; i< delay; i++) {
            System.out.println(message);
        }
    }
    catch(InterruptedException ie){
        System.out.println(ie);
    }
}
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Good morning
Hello
Welcome
Good morning
Hello
All the three threads t1, t2 and t3 have completed execution.

<b>S.No: 24</b>	<b>Exp. Name: <i>Java program that correctly implements the producer consumer problem</i></b>	<b>Date:</b>
-----------------	---	--------------

**Aim:**

Write a Java program that correctly implements the producer-consumer problem using the concept of interthread communication.

**Source Code:**

q2499/ProdCons.java

```
package q2499;
class Q {
    int n;
    boolean valueSet = false;
    synchronized int get() {
        while(!valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Got: " + n);
        valueSet = false;
        notify();
        return n;
    }
    synchronized void put(int n) {
        while(valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.n = n;
        valueSet = true;
        System.out.println("Put: " + n);
        notify();
    }
}
class Producer implements Runnable {
    Q q;
    int n;
    boolean stop=false;
    Producer(Q q,int n) {
        this.q = q;
        this.n=n;
    }
    public void run() {
        for(int i=0;i<n;i++)
        {
            q.put(i);
        }
    }
}
```

```

class Consumer implements Runnable {
    Q q;
    int n;
    boolean stop=false;
    Consumer(Q q,int n) {
        this.q = q;
        this.n=n;
        //new Thread(this, "Consumer").start();
    }
    public void run() {
        for(int i=0;i<n;i++) {
            q.get();
        }
    }
}

class ProdCons {
    public static void main(String args[]) throws Exception{
        Q q = new Q();
        int n=Integer.parseInt(args[0]);
        Producer p=new Producer(q,n);
        Consumer c=new Consumer(q,n);
        Thread t=new Thread(p);
        Thread t1=new Thread(c);
        t.start();
        t1.start();
        System.out.println("Producer-Consumer problem using the concept of Interthread Communication");
    }
}

```

### Execution Results - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
Producer-Consumer problem using the concept of Interthread Communication
Put: 0
Got: 0
Put: 1
Got: 1
Put: 2
Got: 2
Put: 3
Got: 3
Put: 4
Got: 4

Test Case - 2
<b>User Output</b>
Producer-Consumer problem using the concept of Interthread Communication
Put: 0

**Test Case - 2**

Got: 0

Put: 1

Got: 1

Put: 2

Got: 2

Put: 3

Got: 3

Put: 4

Got: 4

Put: 5

Got: 5

Put: 6

Got: 6

Put: 7

Got: 7

**Test Case - 3****User Output**

Producer-Consumer problem using the concept of Interthread Communication

Put: 0

Got: 0

Put: 1

Got: 1

Put: 2

Got: 2

Put: 3

Got: 3

Put: 4

Got: 4

Put: 5

Got: 5



**S.No: 25** Exp. Name: *Iteration on ArrayList***Date:****Aim:**

23. Write a Java program to store student class objects in Array List and sort the ArrayList. Create a class name `ArrayListDemo` with a `main` method. The method takes student names as inputs from the command line arguments. Write to store student class objects in Array List and sort the ArrayList?

**Source Code:**

q24085/ArrayListIterationDemo.java

```
package q24085;
import java.util.*;
    // write your code here
class ArrayListIterationDemo
{
    public static void main(String args[])
    {
        ArrayList<String> obj = new ArrayList<String>();
        for(int i=0;i<args.length;i++)
        {
            obj.add(args[i]);
        }

        System.out.println( obj);
        Collections.sort(obj);
        System.out.println("After sorting : " +obj);
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

[Ram, Ravi, Raj]

After sorting : [Raj, Ram, Ravi]

**Test Case - 2****User Output**

[Lalitha, Lavanya, Ramya]

After sorting : [Lalitha, Lavanya, Ramya]

**Test Case - 3****User Output**

[Suguna, Srujana, Sruthi]

After sorting : [Srujana, Sruthi, Suguna]

**S.No: 26** Exp. Name: *Iteration on ArrayList***Date:****Aim:**

Write a Java program to create HashSet from ArrayList and remove duplicates and display results?

**Source Code:**

ArrayListIterationDemo.java

```
import java.util.*;
class ArrayListIterationDemo
{
    public static void main(String args[])throws Exception
    {
        ArrayList<String> obj=new ArrayList<String>();
        for(int i=0;i<args.length;i++)
        {
            obj.add(args[i]);
        }
        System.out.println(obj);
        HashSet<String> has=new HashSet(obj);

        System.out.println("List after removing duplicate elements:");
        for(String l : has)
            System.out.println(l);
    }
}
```

**Execution Results** - All test cases have succeeded!**Test Case - 1****User Output**

[Ram, Ram, Raj, Ravi]

List after removing duplicate elements:

Ravi

Raj

Ram

**Test Case - 2****User Output**

[suguna, suguna, srujana]

List after removing duplicate elements:

srujana

suguna

S.No: 27

Exp. Name: **Write a java program to traversing elements in descending order in TreeSet.**

Date:

**Aim:**

Write a java program to traversing elements in descending order in TreeSet.

**Source Code:**

HashSetMethodsDemo.java

```
import java.util.*;
class HashSetMethodsDemo
{
    public static void main(String args[])throws Exception
    {
        TreeSet<Object> ints = new TreeSet<Object>();
        for(int i=0;i<args.length;i++)
        {
            ints.add(args[i]);
        }
        TreeSet<Object> intsReverse =
            (TreeSet<Object>)ints.descendingSet();
        System.out.println("Traversing element through Iterator in descending orde
r");
        for(Object ob : intsReverse)
        {
            System.out.println(ob);
        }
    }
}
```

**Execution Results** - All test cases have succeeded!

Test Case - 1
<b>User Output</b>
Traversing element through Iterator in descending order
Yamuna
Rama
Krishna
Ganga



**RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

---

### **Evaluation Procedure for Internal Laboratory Examinations:**

1. Of the 25 marks for internal, 10 marks will be awarded for day-to-day work and 10 marks to be awarded for the Record work and 5 marks to be awarded by conducting an internal laboratory test.
2. Concerned Teachers have to do necessary corrections with explanations.
3. Concerned Lab teachers should enter marks in index page.
4. Internal exam will be conducted by two Staff members.

Dr.K. Subba Reddy

Professor & Head Dept. of CSE.



**RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

---

### **Evaluation Procedure for External Laboratory Examinations:**

1. For Practical subjects there is a continuous evaluation during the semester for 25 Sessional marks and 50 end examination marks.
2. The end examination shall be conducted by the teacher concerned (Internal Examiner) and another External Examiner, recommended by Head of the Department with the approval of principal.

Evaluation procedure for external lab examination:

1. Procedure for the program	----- 20M
2. Execution of the program	----- 15M
3. Viva voce	----- 15M
	-----
Total	50M
	-----

Dr.K. Subba Reddy

Professor & Head Dept. of CSE.