

## SECTION 1

### 1. Write a program to print 'Welcome to Java'.

```
public class Hello{  
    public static void main(String aargs[]){  
        System.out.println("Welcome to Java");  
    }  
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac Hello.java  
  
C:\Users\athul\OneDrive\Desktop\java09>java Hello  
Welcome to Java
```

### 2. WAP to display two numbers received as command line argument, and print its product.

```
import java.io.*;  
import java.lang.*;  
public class NumProd{  
    public static void main(String aargs[]){  
        try{  
            String s;  
            System.out.println("Enter a: ");  
            DataInputStream dis=new DataInputStream(System.in);  
            s=dis.readLine();  
            int a=Integer.parseInt(s);  
            System.out.println("Enter b: ");  
            dis=new DataInputStream(System.in);  
            s=dis.readLine();  
            int b=Integer.parseInt(s);  
            int product=a*b;  
            System.out.println("a: "+a);  
            System.out.println("b: "+b);  
            System.out.println("product of a and b: "+product);  
        }  
        catch(Exception e)  
        {  
            System.out.println(e);  
        }  
    }  
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac NumProd.java  
Note: NumProd.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
  
C:\Users\athul\OneDrive\Desktop\java09>java NumProd  
Enter a:  
3  
Enter b:  
6  
a: 3  
b: 6  
product of a and b: 18
```

### 3. WAP to read two numbers and display the output in the form of 'Sum of 2 and 3 is 5.'

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

```

public class NumAdd{
public static void main(String aargs[]){
try{
String s;
System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int a=Integer.parseInt(s);
System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);
int sum=a+b;
System.out.println("a: "+a);
System.out.println("b: "+b);
System.out.println("Sum of " +a+ " and " +b+ " is " +sum);
}
catch(Exception e)
{
System.out.println(e);
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac NumAdd.java
Note: NumAdd.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java NumAdd
Enter a:
2
Enter b:
5
a: 2
b: 5
Sum of 2 and 5 is 7

```

#### 4.WAP to accept two numbers from the keyboard and swap them.

```

import java.io.*;
public class SwapNum{
public static void main(String args[]){
try{
String s;
System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine()
int a=Integer.parseInt(s);
System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);
System.out.println("Before swapping: ");
System.out.println("a: "+a);
System.out.println("b: "+b);
int t=a;
a=b;
b=t;
System.out.println("After swapping: ");
System.out.println("a: "+a);
}
}
}

```

```
System.out.println("b: "+b);  
}  
catch(Exception e){  
System.out.println(e);  
}  
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac SwapNum.java  
Note: SwapNum.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
  
C:\Users\athul\OneDrive\Desktop\java09>java SwapNum  
Enter a:  
3  
Enter b:  
6  
Before swapping:  
a: 3  
b: 6  
After swapping:  
a: 6  
b: 3
```



## SECTION 2

### 1.WAP to read three numbers and the maximum.

```
import java.io.*;
import java.lang.*;
public class MaxNum{
    public static void main(String args[]){
        try{
            String s;
            System.out.println("Enter a: ");
            DataInputStream dis=new DataInputStream(System.in);
            s=dis.readLine();
            int a=Integer.parseInt(s);
            System.out.println("Enter b: ");
            dis=new DataInputStream(System.in);
            s=dis.readLine();
            int b=Integer.parseInt(s);
            System.out.println("Enter c: ");
            dis=new DataInputStream(System.in);
            s=dis.readLine();
            int c=Integer.parseInt(s);
            int max;
            if(a>=b&&a>=c){
                System.out.println("max: "+a);
            }
            else if(b>=a&&b>=c){
                System.out.println("max: "+b);
            }
            else{
                System.out.println("max: "+c);
            }
        }catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac MaxNum.java
Note: MaxNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java MaxNum
Enter a:
4
Enter b:
6
Enter c:
2
max: 6
```

### 2. Find the minimum of three numbers using a single statement.

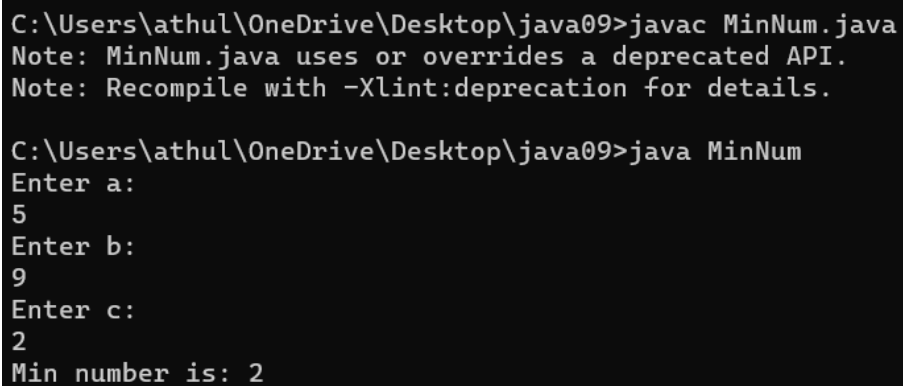
```
import java.io.*;
import java.lang.*;
public class MinNum
{
    public static void main(String args[])
    {
```

```

try
{
String s;
System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int a=Integer.parseInt(s);
System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);

System.out.println("Enter c: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int c=Integer.parseInt(s);
int min=Math.min(Math.min(a,b),c);
System.out.println("Min number is: "+min);
}
catch(Exception e)
{
System.out.println(e);
}
}
}

```



```

C:\Users\athul\OneDrive\Desktop\java09>javac MinNum.java
Note: MinNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java MinNum
Enter a:
5
Enter b:
9
Enter c:
2
Min number is: 2

```

### 3.WAP to search for a given element in an array.

```

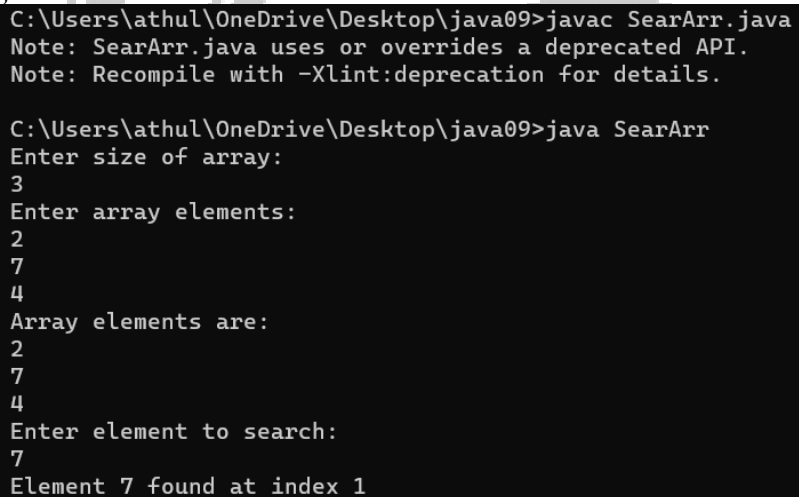
import java.io.*;
public class SearArr {
    public static void main(String args[]) {
        try
        {
            int a[];
            String s;
            int i;
            System.out.println("Enter size of array: ");
            DataInputStream dis = new DataInputStream(System.in);
            s = dis.readLine();
            int n = Integer.parseInt(s)
            a = new int[n];
            System.out.println("Enter array elements: ");
            for (i = 0; i < n; i++) {
                s = dis.readLine();
                a[i] = Integer.parseInt(s);
            }
            System.out.println("Array elements are: ");
            for (i = 0; i < n; i++) {

```

```

System.out.println(a[i]);
}
System.out.println("Enter element to search: ");
s = dis.readLine();
int searchElement = Integer.parseInt(s);
int flag = 0;
for (i = 0; i < n; i++) {
    if (a[i] == searchElement) {
        flag = 1;
        System.out.println("Element " + searchElement + " found at index " + i);
        break;
    }
}
if (flag == 0) {
    System.out.println("Element " + searchElement + " not found in the array.");
}
}
catch (Exception e) {
    System.out.println(e);
}
}
}

```



```

C:\Users\athul\OneDrive\Desktop\java09>javac SearArr.java
Note: SearArr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java SearArr
Enter size of array:
3
Enter array elements:
2
7
4
Array elements are:
2
7
4
Enter element to search:
7
Element 7 found at index 1

```

#### 4. WAP to sort elements in an array in ascending order.

```

import java.io.*;
public class SortArr {
    public static void main(String args[]) {
        try {
            int a[];
            String s;
            System.out.println("Enter size of array: ");
            DataInputStream dis = new DataInputStream(System.in);
            s = dis.readLine();
            int n = Integer.parseInt(s);
            a = new int[n];
            System.out.println("Enter array elements: ");
            for (i = 0; i < n; i++) {
                s = dis.readLine();
                a[i] = Integer.parseInt(s);
            }
            System.out.println("Array elements before sorting: ");
            for (i = 0; i < n; i++) {
                System.out.println(a[i]);
            }
        }
    }
}

```

```

        bubbleSort(a);
        System.out.println("Array elements after sorting in ascending order:");
        for (i = 0; i < n; i++) {
            System.out.println(a[i]);
        }
    } catch (Exception e) {
        System.out.println(e);
    }
}

static void bubbleSort(int[] arr) {
    int n = arr.length;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac SortArr.java
Note: SortArr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java SortArr
Enter size of array:
3
Enter array elements:
5
2
8
Array elements before sorting:
5
2
8
Array elements after sorting in ascending order:
2
5
8

```

**5. Write a program to print the row wise and column wise sum of a 2D array. 1 2 3 | 6**

2 1 1 | 4. . .

3 3 4

import java.io.\*;

```

public class TdArr
{
    public static void main(String[] args) {
        DataInputStream i = new DataInputStream(System.in);
        try {
            int rows, cols;
            int[][] array;
            String s;

            System.out.println("Enter the number of rows for the 2D array:");
            s = i.readLine();
            rows = Integer.parseInt(s);
            System.out.println("Enter the number of columns for the 2D array:");
            s = i.readLine();
            cols = Integer.parseInt(s);
            array = new int[rows][cols];
            for (int j = 0; j < rows; j++) {
                for (int k = 0; k < cols; k++) {
                    System.out.print("Enter element at row " + (j + 1) + ", column " + (k + 1) + ": ");

```

```

        s = i.readLine();
        array[j][k] = Integer.parseInt(s);
    }
}
System.out.println("2D Array:");
for (int j = 0; j < rows; j++) {
    for (int k = 0; k < cols; k++) {
        System.out.print(array[j][k] + " ");
    }
    System.out.println();
}
System.out.println("Row-wise Sum:");
for (int j = 0; j < rows; j++) {
    int rowSum = 0;
    for (int k = 0; k < cols; k++) {
        rowSum += array[j][k];
    }
    System.out.println("Row " + (j + 1) + ": " + rowSum);
}
System.out.println("Column-wise Sum:");
for (int k = 0; k < cols; k++) {
    int colSum = 0;
    for (int j = 0; j < rows; j++) {
        colSum += array[j][k];
    }
    System.out.print("Column " + (k + 1) + ": " + colSum + " ");
}
System.out.println();
} catch (Exception e) {
    System.out.println("Error " + e);
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>java TdArr
Enter the number of rows for the 2D array:
3
Enter the number of columns for the 2D array:
3
Enter element at row 1, column 1: 1
Enter element at row 1, column 2: 2
Enter element at row 1, column 3: 3
Enter element at row 2, column 1: 4
Enter element at row 2, column 2: 5
Enter element at row 2, column 3: 6
Enter element at row 3, column 1: 7
Enter element at row 3, column 2: 8
Enter element at row 3, column 3: 9
2D Array:
1 2 3
4 5 6
7 8 9
Row-wise Sum:
Row 1: 6
Row 2: 15
Row 3: 24
Column-wise Sum:
Column 1: 12 Column 2: 15 Column 3: 18

```



## SECTION 3

**1. WAP with two functions to check for an integer palindrome. (Function1 should reverse the integer. Function2 should return 1,if it is a palindrome or else 0.)**

```
import java.io.*;
public class PalindromeChecker {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            String input = dis.readLine();
            int num = Integer.parseInt(input);
            System.out.println("Original number: " + num);
            if (isPalindrome(num)) {
                System.out.println("The number is a palindrome.");
            } else {
                System.out.println("The number is not a palindrome.");
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static int reverseInteger(int num) {
        int reversed = 0;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        return reversed;
    }
    public static boolean isPalindrome(int num) {
        return num == reverseInteger(num);
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac PalindromeChecker.java
Note: PalindromeChecker.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java PalindromeChecker
Enter a number: 232
Original number: 232
The number is a palindrome.
```

**2. WAP to display numbers from m to n using single while loop.**

```
import java.io.*;
public class NumbersWhile {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the starting number (m): ");
            String mStr = dis.readLine();
            int m = Integer.parseInt(mStr);
            System.out.print("Enter the ending number (n): ");
```

```

String nStr = dis.readLine();
int n = Integer.parseInt(nStr);
System.out.println("Numbers from " + m + " to " + n + ":");
while (m <= n) {
    System.out.println(m);
    m++;
}
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac NumbersWhile.java
Note: NumbersWhile.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java NumbersWhile
Enter the starting number (m): 2
Enter the ending number (n): 5
Numbers from 2 to 5:
2
3
4
5

```

### 3. WAP to find the sum of the series $1+(1+2)+(1+2+3)+\dots+(1+2+3+\dots+n)$ using a single while loop.

```

import java.io.*;
public class SumSeries {
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the value of n: ");
            int n = Integer.parseInt(dis.readLine());
            int sum = 0;
            int innerSum = 0;
            int i = 1;
            while (i <= n) {
                innerSum += i;
                sum += innerSum;
                i++;
            }
            System.out.println("Sum of the series is: " + sum);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac SumSeries.java
Note: SumSeries.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java SumSeries
Enter the value of n: 4
Sum of the series is: 20

```

### 4. WAP to find the sum of $1+2/2!+3/3!+4/4!+\dots+n/n!$ using a single for loop.

```

import java.io.*;

```

```

public class FactorialSum {
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the value of n: ");
            int n = Integer.parseInt(dis.readLine());
            double sum = 0;
            double factorial = 1;
            for (int i = 1; i <= n; i++) {
                factorial *= i;
                sum += i / factorial;
            }
            System.out.println("Sum of the series is: " + sum);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac FactorialSum.java
Note: FactorialSum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java FactorialSum
Enter the value of n: 3
Sum of the series is: 2.5

```

##### 5. WAP to calculate area of a circle (functions with no argument and no return type.)

```

import java.io.*;
public class CircleArea {
    static void calculateArea() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.println("Enter the radius of the circle: ");
            double radius = Double.parseDouble(dis.readLine());
            double area = Math.PI * radius * radius;
            System.out.println("Area of the circle: " + area);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static void main(String args[]) {
        calculateArea();
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac CircleArea.java
Note: CircleArea.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java CircleArea
Enter the radius of the circle:
3
Area of the circle: 28.274333882308138

```

#### 6. WAP to reverse a number (functions with argument and no return type.)

```
import java.io.*;

public class ReverseNumber {
    void reverseNumber(int number) {
        int reversedNumber = 0;
        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number /= 10;
        }
        System.out.println("Reversed Number: " + reversedNumber);
    }

    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());

            ReverseNumber reverseNumberObj = new ReverseNumber();
            reverseNumberObj.reverseNumber(inputNumber);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac ReverseNumber.java
Note: ReverseNumber.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java ReverseNumber
Enter a number: 123
Reversed Number: 321
```

#### 7. WAP to calculate sum of digits of a number (functions with argument and return type.)

```
import java.io.*;

public class SumOfDigits {
    static int calculateSumOfDigits(int number) {
        int sum = 0;
        while (number != 0) {
            int digit = number % 10;
            sum += digit;
            number /= 10;
        }
        return sum;
    }

    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());
            int sum = calculateSumOfDigits(inputNumber);
            System.out.println("Sum of digits: " + sum);
        } catch (Exception e) {
        }
    }
}
```

```

        System.out.println(e);
    }
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac ReverseNumber.java
Note: ReverseNumber.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

```

```

C:\Users\athul\OneDrive\Desktop\java09>java ReverseNumber
Enter a number: 123
Reversed Number: 321

```

#### 8. WAP to calculate sum of n even numbers (functions with no argument and return type.)

```

import java.io.*;
public class SumOfEven {
    static void calculateSumOfEvenNumbers() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the value of n: ");
            int n = Integer.parseInt(dis.readLine());
            int sum = 0;
            int count = 0;
            int number = 2; // Starting from the first even number
            while (count < n) {
                sum += number;
                number += 2; // Move to the next even number
                count++;
            }
            System.out.println("Sum of the first " + n + " even numbers: " + sum);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static void main(String args[]) {
        calculateSumOfEvenNumbers();
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac SumOfEven.java
Note: SumOfEven.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

```

```

C:\Users\athul\OneDrive\Desktop\java09>java SumOfEven
Enter the value of n: 3
Sum of the first 3 even numbers: 12

```

## SECTION 4

**1.WAP with nested functions to find the maximum of three numbers. Function1 should take in two arguments and find the maximum. Function2 should take in the third number and the maximum from function1 to find the maximum.)**

```
import java.io.*;
public class MaxOfThreeNumbers {
    static int findMax(int num1, int num2) {
        return (num1 > num2) ? num1 : num2;
    }
    static int findMaxOfThree(int num1, int num2, int num3) {
        int maxOfFirstTwo = findMax(num1, num2);
        return findMax(maxOfFirstTwo, num3);
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the first number: ");
            int num1 = Integer.parseInt(dis.readLine());
            System.out.print("Enter the second number: ");
            int num2 = Integer.parseInt(dis.readLine());
            System.out.print("Enter the third number: ");
            int num3 = Integer.parseInt(dis.readLine());
            int max = findMaxOfThree(num1, num2, num3);
            System.out.println("Maximum of the three numbers: " + max);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac MaxOfThreeNumbers.java
Note: MaxOfThreeNumbers.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java MaxOfThreeNumbers
Enter the first number: 3
Enter the second number: 2
Enter the third number: 5
Maximum of the three numbers: 5
```

**2. WAP to find the factorial of n, using recursion.**

```
import java.io.*;
public class FactorialWithRecursion {
    static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number : ");
            int inputNumber = Integer.parseInt(dis.readLine());
            int result = factorial(inputNumber);
            System.out.println("Factorial of " + inputNumber + " is: " + result);
        }
    }
}
```

```

    } catch (Exception e) {
        System.out.println(e);
    }
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac FactorialWithRecursion.java
Note: FactorialWithRecursion.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java FactorialWithRecursion
Enter a number : 3
Factorial of 3 is: 6

```

### 3. WAP to display numbers from n to 1 and vice versa, using recursion.

```

import java.io.*;
public class DisplayNumbersRecursion {
    static void displayDescending(int n) {
        if (n >= 1) {
            System.out.print(n + " ");
            displayDescending(n - 1);
        }
    }
    static void displayAscending(int n) {
        if (n >= 1) {
            displayAscending(n - 1);
            System.out.print(n + " ");
        }
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());
            System.out.print("Descending Order: ");
            displayDescending(inputNumber);
            System.out.println(); // Move to the next line
            System.out.print("Ascending Order: ");
            displayAscending(inputNumber);
            System.out.println(); // Move to the next line
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac DisplayNumbersRecursion.java
Note: DisplayNumbersRecursion.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java DisplayNumbersRecursion
Enter a number: 4
Descending Order: 4 3 2 1
Ascending Order: 1 2 3 4

```

### 4. Using constructors, implement the operations of a queue.

```

import java.io.*;

```

```

class Queue {
    private static final int MAX_SIZE = 5; // Maximum size of the queue
    private int[] array;
    private int front, rear, size;
    public Queue() {
        array = new int[MAX_SIZE];
        front = rear = -1;
        size = 0;
    }
    public boolean isEmpty() {
        return size == 0;
    }
    public boolean isFull() {
        return size == MAX_SIZE;
    }
    public void enqueue(int element) {
        if (isFull()) {
            System.out.println("Queue is full. Cannot enqueue.");
        } else {
            if (isEmpty()) {
                front = 0;
            }
            rear = (rear + 1) % MAX_SIZE;
            array[rear] = element;
            size++;
            System.out.println("Enqueued: " + element);
        }
    }
    public void dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty. Cannot dequeue.");
        } else {
            int removedElement = array[front];
            if (front == rear) {
                front = rear = -1;
            } else {
                front = (front + 1) % MAX_SIZE;
            }
            size--;
            System.out.println("Dequeued: " + removedElement);
        }
    }
    public void display() {
        if (isEmpty()) {
            System.out.println("Queue is empty.");
        } else {
            System.out.print("Queue: ");
            int i = front;
            do {
                System.out.print(array[i] + " ");
                i = (i + 1) % MAX_SIZE;
            } while (i != (rear + 1) % MAX_SIZE);
            System.out.println();
        }
    }
}

public class QueueImplementation {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);

```



```

Queue myQueue = new Queue();
while (true) {
    System.out.println("\nQueue Operations:");
    System.out.println("1. Enqueue");
    System.out.println("2. Dequeue");
    System.out.println("3. Display");
    System.out.println("4. Exit");
    System.out.print("Enter your choice: ");
    int choice = Integer.parseInt(dis.readLine());
    switch (choice) {
        case 1:
            System.out.print("Enter the element to enqueue: ");
            int element = Integer.parseInt(dis.readLine());
            myQueue.enqueue(element);
            break;

        case 2:
            myQueue.dequeue();
            break;

        case 3:
            myQueue.display();
            break;

        case 4:
            System.out.println("Exiting the program.");
            System.exit(0);
            break;

        default:
            System.out.println("Invalid choice. Please enter a valid option.");
    }
}
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac QueueImplementation.java
Note: QueueImplementation.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

```

```

C:\Users\athul\OneDrive\Desktop\java09>java QueueImplementation

```

```

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter the element to enqueue: 2
Enqueued: 2

```

```

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
Queue: 2

```

```

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 2
Dequeued: 2

```

**5. Create a class “weight” having a kg and gm part. Provide functions for read, display, add and subtracting two weights.**

```
import java.util.Scanner;

class Weight {
    private int kg;
    private int gm;

    public Weight() {
        kg = 0;
        gm = 0;
    }

    public Weight(int kg, int gm) {
        this.kg = kg;
        this.gm = gm;
    }

    public void readWeight() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the weight in kg: ");
        kg = sc.nextInt();
        System.out.print("Enter the weight in gm: ");
        gm = sc.nextInt();
    }

    public void displayWeight() {
        System.out.println("Weight: " + kg + " kg " + gm + " gm");
    }

    public Weight addWeight(Weight w) {
        Weight sum = new Weight();
        sum.kg = kg + w.kg;
        sum.gm = gm + w.gm;
        if (sum.gm >= 1000) {
            sum.kg += sum.gm / 1000;
            sum.gm %= 1000;
        }
        return sum;
    }

    public Weight subtractWeight(Weight w) {
        Weight diff = new Weight();
        if (kg < w.kg || (kg == w.kg && gm < w.gm)) {
            System.out.println("Error: Cannot subtract a larger weight from a smaller weight");
            return diff;
        }
        diff.kg = kg - w.kg;
        if (gm < w.gm) {
            diff.kg--;
            diff.gm = gm + 1000 - w.gm;
        } else {
            diff.gm = gm - w.gm;
        }
        return diff;
    }
}
```

```

public class WeightOperations {
    public static void main(String[] args) {
        Weight w1 = new Weight();
        w1.readWeight();
        w1.displayWeight();

        Weight w2 = new Weight();
        w2.readWeight();
        w2.displayWeight();

        Weight sum = w1.addWeight(w2);
        System.out.print("Sum of weights: ");
        sum.displayWeight();

        Weight diff = w1.subtractWeight(w2);
        System.out.print("Difference of weights: ");
        diff.displayWeight();
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac ComplexOperations.java
Note: ComplexOperations.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java ComplexOperations
Enter details for Complex Number 1:
Enter the real part: 1
Enter the imaginary part: 3

Enter details for Complex Number 2:
Enter the real part: 5
Enter the imaginary part: 2

Entered Complex Numbers:
Complex Number: 1.0 + 3.0i
Complex Number: 5.0 + 2.0i

Sum of Complex Numbers:
Complex Number: 6.0 + 5.0i

Product of Complex Numbers:
Complex Number: -1.0 + 17.0i

```

#### 6. WAP to display even numbers upto 'n' using a static function.

```

import java.io.*;
public class DisplayEvenNumber {
    static void displayEvenNumbers(int n) {
        System.out.println("Even numbers up to " + n + ":");
        for (int i = 2; i <= n; i += 2) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number 'n': ");
            int inputNumber = Integer.parseInt(dis.readLine());
            displayEvenNumbers(inputNumber);
        } catch (Exception e) {

```

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

```
C:\Users\athul\OneDrive\Desktop\java09>javac DisplayEvenNumber.java  
Note: DisplayEvenNumber.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java DisplayEvenNumber  
Enter a number 'n': 6  
Even numbers up to 6:  
2 4 6
```



## SECTION 5

**1. WAP (menu driven) to demonstrate method overriding in java, by displaying details of a student, and a teacher.**

```
import java.io.*;
class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
class Student extends Person {
    private int rollNumber;
    public Student(String name, int age, int rollNumber) {
        super(name, age);
        this.rollNumber = rollNumber;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Roll Number: " + rollNumber);
    }
}
class Teacher extends Person {
    private String subject;
    public Teacher(String name, int age, String subject) {
        super(name, age);
        this.subject = subject;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Subject: " + subject);
    }
}
public class MethodOverridingDemo {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            while (true) {
                System.out.println("\nMenu:");
                System.out.println("1. Display Student Details");
                System.out.println("2. Display Teacher Details");
                System.out.println("3. Exit");
                System.out.print("Enter your choice: ");
                int choice = Integer.parseInt(dis.readLine());
                switch (choice) {
                    case 1:
                        Student student = new Student("James", 20, 43);
                        student.displayDetails();
                        break;
                    case 2:
                        Teacher teacher = new Teacher("Bindhu", 35, "Mathematics");
                        teacher.displayDetails();
                        break;
                    case 3:
```

```

        System.out.println("Exiting the program.");
        System.exit(0);
        break;
    default:
        System.out.println("Invalid choice. Please enter a valid option.");
    }
}
} catch (Exception e) {
    System.out.println(e);
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac MethodOverridingDemo.java
Note: MethodOverridingDemo.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java MethodOverridingDemo

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 1
Name: James
Age: 20
Roll Number: 43

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 2
Name: Bindhu
Age: 35
Subject: Mathematics

```

**2. Create a class for employee having eno,ename and esal as data members. Provide functions for reading and displaying employee details. (Accept information of n employees in the main function, display the same and search for an emp (using eno)).**

```

import java.io.*;
class Employee {
    private int eno;
    private String ename;
    private double esal;
    public void readDetails() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter Employee Number : ");
            eno = Integer.parseInt(dis.readLine());
            System.out.print("Enter Employee Name : ");
            ename = dis.readLine();
            System.out.print("Enter Employee Salary : ");
            esal = Double.parseDouble(dis.readLine());
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public void displayDetails() {
        System.out.println("Employee Details:");
        System.out.println("Employee Number : " + eno);
        System.out.println("Employee Name : " + ename);
        System.out.println("Employee Salary : " + esal);
    }
}

```

```

    }
    public boolean searchEmployee(int searchEno) {
        return eno == searchEno;
    }
}
public class EmployeeDetails {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the number of employees (n): ");
            int n = Integer.parseInt(dis.readLine());
            Employee[] employees = new Employee[n];
            for (int i = 0; i < n; i++) {
                employees[i] = new Employee();
                System.out.println("\nEnter details for Employee " + (i + 1) + ":");
                employees[i].readDetails();
            }
            System.out.println("\nDetails of all Employees:");
            for (int i = 0; i < n; i++) {
                employees[i].displayDetails();
                System.out.println();
            }
            System.out.print("Enter Employee Number to search: ");
            int searchEno = Integer.parseInt(dis.readLine());
            boolean found = false;

            for (int i = 0; i < n; i++) {
                if (employees[i].searchEmployee(searchEno)) {
                    System.out.println("Employee found!");
                    employees[i].displayDetails();
                    found = true;
                    break;
                }
            }
            if (!found) {
                System.out.println("Employee not found.");
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac EmployeeDetails.java
Note: EmployeeDetails.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java EmployeeDetails
Enter the number of employees (n): 2

Enter details for Employee 1:
Enter Employee Number : 01
Enter Employee Name : Aryan
Enter Employee Salary : 2000

Enter details for Employee 2:
Enter Employee Number : 02
Enter Employee Name : Madhav
Enter Employee Salary : 2000

Details of all Employees:
Employee Details:
Employee Number : 1
Employee Name : Aryan
Employee Salary : 2000.0

Employee Details:
Employee Number : 2
Employee Name : Madhav
Employee Salary : 2000.0

Enter Employee Number to search: 02
Employee found!
Employee Details:
Employee Number : 2
Employee Name : Madhav
Employee Salary : 2000.0

```

## SECTION 6

### 1. Program to implement run time polymorphism in Java using interface, wrt calculating area of a triangle.

```
import java.io.*;
interface Shape {
    double calculateArea();
}
class Triangle implements Shape {
    private double base;
    private double height;
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    public double calculateArea() {
        return 0.5 * base * height;
    }
}
public class AreaTriangle {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the base of the triangle: ");
            double base = Double.parseDouble(dis.readLine());
            System.out.print("Enter the height of the triangle: ");
            double height = Double.parseDouble(dis.readLine());
            Triangle triangle = new Triangle(base, height);
            System.out.println("Area of the triangle: " + triangle.calculateArea());
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac AreaTriangle.java
Note: AreaTriangle.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java AreaTriangle
Enter the base of the triangle: 3
Enter the height of the triangle: 6
Area of the triangle: 9.0
```

### 2. Create an interface Shape having two prototypes disp() and calc(), to display the shape and calculate volume respectively. Create two classes circle and rectangle which implements the above interface. In the main function create a reference of Shape depending on the user-choice.

```
import java.io.*;
interface Shape {
    void disp();
    void calc();
}
class Circle implements Shape {
    private double radius;
    public Circle(double radius) {
```



```

        this.radius = radius;
    }
    public void disp() {
        System.out.println("Displaying Circle");
    }
    public void calc() {
        double area = Math.PI * radius * radius;
        System.out.println("Calculating Area of Circle: " + area);
    }
}

class Rectangle implements Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public void disp() {
        System.out.println("Displaying Rectangle");
    }
    public void calc() {
        double area = length * width;
        System.out.println("Calculating Area of Rectangle: " + area);
    }
}

public class InterfaceShape {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.println("Choose a shape:");
            System.out.println("1. Circle");
            System.out.println("2. Rectangle");
            System.out.print("Enter your choice (1 or 2): ");
            int choice = Integer.parseInt(dis.readLine());
            Shape shape;
            if (choice == 1) {
                System.out.print("Enter the radius of the circle: ");
                double radius = Double.parseDouble(dis.readLine());
                shape = new Circle(radius);
            } else if (choice == 2) {
                System.out.print("Enter the length of the rectangle: ");
                double length = Double.parseDouble(dis.readLine());
                System.out.print("Enter the width of the rectangle: ");
                double width = Double.parseDouble(dis.readLine());
                shape = new Rectangle(length, width);
            } else {
                System.out.println("Invalid choice. Exiting the program.");
                return;
            }
            shape.disp();
            shape.calc();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac InterfaceShape.java
Note: InterfaceShape.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java InterfaceShape
Choose a shape:
1. Circle
2. Rectangle
Enter your choice (1 or 2): 1
Enter the radius of the circle: 3
Displaying Circle
Calculating Area of Circle: 28.274333882308138
```

### 3. WAP to implement a function using call by value to swap two float numbers.

```
import java.io.*;
public class swapFloat
{
    static void swap(float a, float b) {
        System.out.println("Before swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        float temp = a;
        a = b;
        b = temp;
        System.out.println("After swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the first float number (a): ");
            float a = Float.parseFloat(dis.readLine());
            System.out.print("Enter the second float number (b): ");
            float b = Float.parseFloat(dis.readLine());
            swap(a, b);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac swapFloat.java
Note: swapFloat.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java swapFloat
Enter the first float number (a): 21.33
Enter the second float number (b): 36.89
Before swapping:
a = 21.33
b = 36.89
After swapping:
a = 36.89
b = 21.33
```

**4. WAP to implement a function using call by reference to find the square root of a given number.**

```
import java.io.*;
public class SquareRoot {
    static void findSquareRoot(double[] number) {
        if (number[0] >= 0) {
            number[0] = Math.sqrt(number[0]);
        } else {
            System.out.println("Cannot find square root of a negative number.");
        }
    }
}
public static void main(String[] args) {
    try {
        DataInputStream dis = new DataInputStream(System.in);
        System.out.print("Enter a number: ");
        double[] inputNumber = { Double.parseDouble(dis.readLine()) };
        findSquareRoot(inputNumber);
        System.out.println("Square root: " + inputNumber[0]);
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac SquareRoot.java
Note: SquareRoot.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java SquareRoot
Enter a number: 43
Square root: 6.557438524302
```

## SECTION 7

1.Create a class for Cstring having a string data member and provide functions for read , display, compare(return Boolean value),add and concatenate.

```
import java.util.Scanner;
public class CString {
    public String str;
    public CString(String str) {
        this.str = str;
    }
    public void read() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a C-style string: ");
        str = scanner.nextLine();
    }
    public void display() {
        System.out.println("C-style string: " + str);
    }
    public boolean compare(CString other) {
        return this.str.equals(other.str);
    }
    public void add(char ch) {
        str += ch;
    }
    public void concatenate(CString other) {
        str += other.str;
    }
    public static void main(String[] args) {
        CString cString1 = new CString("");
        CString cString2 = new CString("");
        cString1.read();
        cString2.read();
        cString1.display();
        cString2.display();
        if (cString1.compare(cString2)) {
            System.out.println("Strings are equal.");
        } else {
            System.out.println("Strings are not equal.");
        }
        cString1.add('A');
        cString1.display();
        cString1.concatenate(cString2);
        cString1.display();
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac CString.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java CString
```

```
Enter a C-style string: java
Enter a C-style string: program
C-style string: java
C-style string: program
Strings are not equal.
C-style string: javaA
C-style string: javaAprogram
```

**2. Write a program to implement object cloning for the class Distance which has inch and feet as data members.**

```
import java.io.*;
import java.lang.*;
class Dist implements Cloneable{

    Double inch,feet;
    Dist(Double a,Double b){
        inch=a;
        feet=b;
    }

    public Object clone(){
        try{
            return super.clone();
        }
        catch(CloneNotSupportedException c){
            System.out.println("error : "+c);
        }
    }
    return this;
}
}
}
class cloneDemo{
public static void main(String args[]){
    Dist obj1= new Dist(9.1,43.0);
    Dist obj2=(Dist)obj1.clone();
    System.out.println("inch= "+obj2.inch);
    System.out.println("feet =" +obj2.feet);
}
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac cloneDemo.java

C:\Users\athul\OneDrive\Desktop\java09>java cloneDemo
inch= 9.1
feet =43.0
```

**3. Write a program to create a menu driven program for performing the following operations.**

- **Length of a given string**
- **Compare for equality**
- **Extract a substring from a string.**
- **Convert to uppercase and lowercase**

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

class Cstr{
    static void length(String name1,String name2){
        System.out.println("Length of the first string is : "+name1.length());
        System.out.println("Length of the second string is : "+name2.length());
    }
    static void compare(String name1,String name2){
        System.out.println(name1==name2);
    }
}
```

```

static void substr(String name1){
    System.out.println("Substring of "+ name1+" is "+name1.substring(3));
}
static void lowercase(String name1,String name2){
    System.out.println(name1.toLowerCase());
    System.out.println(name2.toLowerCase());
}
public static void main(String args[]){
    try{
        String name1,name2;
        int c,ch=1;

        DataInputStream dir=new DataInputStream(System.in);
        Scanner console=new Scanner(System.in);

        System.out.println("first string : ");
        name1 = console.nextLine();

        System.out.println("second string : ");
        name2 = console.nextLine();

        do{
            System.out.println("1.Length\n2.Compare\n3.Substring\n4.Lowercase\n");
            c=Integer.parseInt(dir.readLine());

            switch(c){

                case 1:
                    length(name1,name2);break;
                case 2:
                    compare(name1,name2);break;
                case 3:
                    substr(name1);break;
                case 4:
                    lowercase(name1,name2);break;

            }
            System.out.println("Do you want to continue (1/0) ");
            ch=Integer.parseInt(dir.readLine());
        }while(ch==1);

    }
    catch(Exception e){
        System.out.println("error "+e);
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac Cstr.java
Note: Cstr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java Cstr
first string :
Athu
second string :
Athulya
1.Length
2.Compare
3.Substring
4.Lowercase

1
Length of the first string is : 4
Length of the second string is : 7
Do you want to continue (1/0)

2
false

3
Substring of Athu is u

4
athu
athulya
Do you want to continue (1/0)

```

#### 4. Write a program to reverse a string.

```

import java.util.Scanner;
public class ReverseString {
    public static String reverseString(String input) {
        char[] charArray = input.toCharArray();
        int start = 0;
        int end = charArray.length - 1;
        while (start < end) {
            char temp = charArray[start];
            charArray[start] = charArray[end];
            charArray[end] = temp;
            start++;
            end--;
        }
        return new String(charArray);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();
        String reversedString = reverseString(inputString);
        System.out.println("Reversed string: " + reversedString);
    }
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac ReverseString.java

C:\Users\athul\OneDrive\Desktop\java09>java ReverseString
Enter a string: java
Reversed string: avaj
```

**5. Write a program to calculate the prime factors of a given number, using packages.**

```
// PrimeFactorsCalculator.java
package mathutils;
import java.util.ArrayList;
import java.util.List;
public class PrimeFactorsCalculator {
    public static List<Integer> calculatePrimeFactors(int number) {
        List<Integer> primeFactors = new ArrayList<Integer>();
        for (int i = 2; i <= number; i++) {
            while (number % i == 0) {
                primeFactors.add(i);
                number /= i;
            }
        }
        return primeFactors;
    }
}

// PrimeFactorsApp.java
import mathutils.PrimeFactorsCalculator;
import java.util.List;
import java.util.Scanner;
public class PrimeFactorsApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to find its prime factors: ");
        int number = scanner.nextInt();
        List<Integer> primeFactors = PrimeFactorsCalculator.calculatePrimeFactors(number);
        System.out.println("Prime factors of " + number + ": " + primeFactors);
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09\packageprgrm>javac PrimeFactorsApp.java

C:\Users\athul\OneDrive\Desktop\java09\packageprgrm>java PrimeFactorsApp
Enter a number to find its prime factors: 40
Prime factors of 40: [2, 2, 2, 5]
```

**6. Write a program to calculate the value of  $nCr$  for given value of  $n$  &  $r$ , using packages.**

```
// mathutils/CombinationCalculator.java
package mathutils;
public class CombinationCalculator {
    private static long factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }
    public static long calculateCombination(int n, int r) {
        if (n < r) {
            return 0; // Invalid input, as n should be greater than or equal to r
        }
    }
}
```



```
    }  
    long numerator = factorial(n);  
    long denominator = factorial(r) * factorial(n - r);  
    return numerator / denominator;  
  }  
}
```

```
// CombinationApp.java  
import mathutils.CombinationCalculator;  
import java.util.Scanner;  
public class CombinationApp {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the value of n: ");  
        int n = scanner.nextInt();  
        System.out.print("Enter the value of r: ");  
        int r = scanner.nextInt();  
        long result = CombinationCalculator.calculateCombination(n, r);  
        System.out.println("Value of " + n + "C" + r + ": " + result);  
    }  
}
```

```
C:\Users\athul\OneDrive\Desktop\java09\packageprgrm>javac CombinationApp.java
```

```
C:\Users\athul\OneDrive\Desktop\java09\packageprgrm>java CombinationApp  
Enter the value of n: 6  
Enter the value of r: 4  
Value of 6C4: 15
```

## SECTION 8

### 1. Read numbers into an array. Perform validations using multiple catch statements / predefined Exceptions.

```
import java.io.DataInputStream;
import java.io.IOException;
public class NumberArrayValidation {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the size of the array: ");
            int size = Integer.parseInt(dis.readLine());

            if (size <= 0) {
                throw new NegativeArraySizeException("Array size must be positive.");
            }
            int[] numbers = new int[size];
            System.out.println("Enter " + size + " numbers separated by spaces:");

            String[] inputNumbers = dis.readLine().split(" ");
            if (inputNumbers.length != size) {
                throw new IOException("Number of input numbers does not match the specified array size.");
            }

            for (int i = 0; i < size; i++) {
                numbers[i] = Integer.parseInt(inputNumbers[i]);
            }

            System.out.println("The numbers you entered are:");
            for (int number : numbers) {
                System.out.println(number);
            }

            dis.close();
        } catch (IOException e) {
            System.out.println("IO Exception occurred.");
            e.printStackTrace();
        } catch (NumberFormatException e) {
            System.out.println("Invalid input format. Please enter numbers separated by spaces.");
        } catch (NegativeArraySizeException e) {
            System.out.println("Negative array size entered. Please enter a positive size.");
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac NumberArrayValidation.java
Note: NumberArrayValidation.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
```

```
C:\Users\athul\OneDrive\Desktop\java09>java NumberArrayValidation
Enter the size of the array: 4
Enter 4 numbers separated by spaces:
7 3 7 2 8
IO Exception occurred.
java.io.IOException: Number of input numbers does not match the specified array size.
    at NumberArrayValidation.main(NumberArrayValidation.java:21)
```

### 2. Write a program to implement a user defined Exception, which will throw an Exception when a given number is prime.

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

```

class PrimeNumberException extends Exception {
    public PrimeNumberException(String message) {
        super(message);
    }
}

public class PrimeChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        try {
            System.out.print("Enter a number to check if it's prime: ");
            int number = scanner.nextInt();
            if (isPrime(number)) {
                throw new PrimeNumberException(number + " is a prime number.");
            } else {
                System.out.println(number + " is not a prime number.");
            }
        } catch (PrimeNumberException e) {
            System.out.println("Exception: " + e.getMessage());
        } catch (Exception e) {
            System.out.println("An error occurred: " + e.getMessage());
        } finally {
            scanner.close(); // Close the scanner to prevent resource leak
        }
    }

    public static boolean isPrime(int number) {
        if (number <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(number); i++) {
            if (number % i == 0) {
                return false;
            }
        }
        return true;
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac PrimeChecker.java

C:\Users\athul\OneDrive\Desktop\java09>java PrimeChecker
Enter a number to check if it's prime: 5
Exception: 5 is a prime number.

```

### 3. Write a program to implement throw and finally.

```

public class ThrowFinallyExample {

    public static void main(String[] args) {
        try {
            int result = divide(10, 0);
            System.out.println("Result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("ArithmeticException caught: Cannot divide by zero.");
        } finally {
            System.out.println("Finally block executed.");
        }
    }

    public static int divide(int num1, int num2) {

```

```
if (num2 == 0) {  
    throw new ArithmeticException("Cannot divide by zero.");  
}  
return num1 / num2;  
  
}}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac ThrowFinallyExample.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java ThrowFinallyExample  
ArithmeticException caught: Cannot divide by zero.  
Finally block executed.
```

## SECTION 9

### 1. Write a program to create multiple threads by extending the Thread class.

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + i);
            try {
                Thread.sleep(1000); // Sleep for 1 second
            } catch (InterruptedException e) {
                System.out.println(e);
            }
        }
    }
}

public class MultiThread {
    public static void main(String[] args) {
        MyThread thread1 = new MyThread();
        MyThread thread2 = new MyThread();
        MyThread thread3 = new MyThread();

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");
        thread3.setName("Thread 3");

        thread1.start();
        thread2.start();
        thread3.start();
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac MultiThread.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java MultiThread
```

```
Thread 1: 1
Thread 2: 1
Thread 3: 1
Thread 2: 2
Thread 3: 2
Thread 1: 2
Thread 1: 3
Thread 2: 3
Thread 3: 3
Thread 1: 4
Thread 2: 4
Thread 3: 4
Thread 1: 5
Thread 3: 5
Thread 2: 5
```

### 2. Write a program to implement threads by implementing the Runnable interface.

```
class MyRunnable implements Runnable {
    public void run() {
        for (int i = 1; i <= 5; i++) {
            System.out.println(Thread.currentThread().getName() + ": " + i);
            try {
```

```

        Thread.sleep(1000); // Sleep for 1 second
    } catch (InterruptedException e) {
        System.out.println(e);
    }
}
}
}

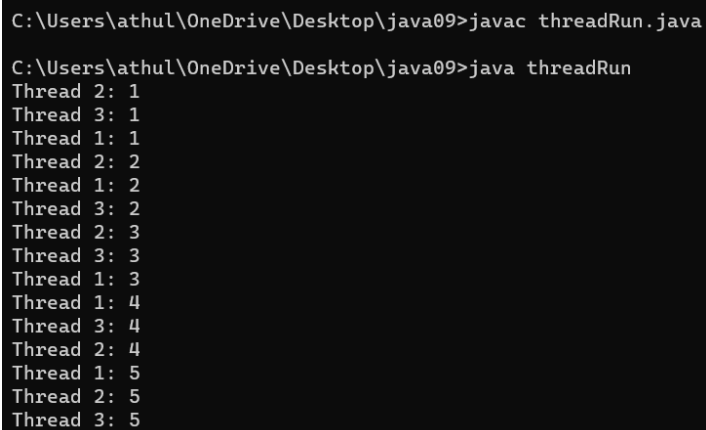
public class threadRun {
    public static void main(String[] args) {
        MyRunnable myRunnable = new MyRunnable();

        Thread thread1 = new Thread(myRunnable);
        Thread thread2 = new Thread(myRunnable);
        Thread thread3 = new Thread(myRunnable);

        thread1.setName("Thread 1");
        thread2.setName("Thread 2");
        thread3.setName("Thread 3");

        thread1.start();
        thread2.start();
        thread3.start();
    }
}

```



```

C:\Users\athul\OneDrive\Desktop\java09>javac threadRun.java

C:\Users\athul\OneDrive\Desktop\java09>java threadRun
Thread 2: 1
Thread 3: 1
Thread 1: 1
Thread 2: 2
Thread 1: 2
Thread 3: 2
Thread 2: 3
Thread 3: 3
Thread 1: 3
Thread 1: 4
Thread 3: 4
Thread 2: 4
Thread 1: 5
Thread 2: 5
Thread 3: 5

```

### 3. Write a program to implement Synchronization using inter-thread communication.

```

public class InterThreadCommunicationExample {
    public static void main(String[] args) {
        SharedResource resource = new SharedResource();

        Thread producerThread = new Thread(new Producer(resource));
        Thread consumerThread = new Thread(new Consumer(resource));

        producerThread.start();
        consumerThread.start();
    }
}

class SharedResource {
    private int data;
    private boolean available = false;

    public synchronized void produce(int newData) {
        while (available) {

```

```

        try {
            wait();
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
    data = newData;
    available = true;
    System.out.println("Produced: " + data);
    notify();
}
public synchronized int consume() {
    while (!available) {
        try {
            wait();
        } catch (InterruptedException e) {
            Thread.currentThread().interrupt();
        }
    }
    available = false;
    System.out.println("Consumed: " + data);
    notify();
    return data;
}
}
class Producer implements Runnable {
    private final SharedResource resource;

    public Producer(SharedResource resource) {
        this.resource = resource;
    }
    public void run() {
        for (int i = 0; i < 5; i++) {
            resource.produce(i);
            try {
                Thread.sleep((long) (Math.random() * 1000));
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
class Consumer implements Runnable {
    private final SharedResource resource;

    public Consumer(SharedResource resource) {
        this.resource = resource;
    }
    public void run() {
        for (int i = 0; i < 5; i++) {
            resource.consume();
            try {
                Thread.sleep((long) (Math.random() * 1000)); // simulate some work
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac InterThreadCommunicationExample.java

C:\Users\athul\OneDrive\Desktop\java09>java InterThreadCommunicationExample
Produced: 0
Consumed: 0
Produced: 1
Consumed: 1
Produced: 2
Consumed: 2
Produced: 3
Consumed: 3
Produced: 4
Consumed: 4
```

#### 4. Implement the Producer- Consumer Problem, using Threads.

```
import java.lang.Thread.*;

class Q {
    int i;
    int maxSize;
    int producedCount = 0;
    boolean produced = false;

    Q(int maxSize) {
        this.maxSize = maxSize;
    }

    synchronized void produce(int x) {
        if (produced) {
            try {
                wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        i = x;
        produced = true;
        System.out.println("Produced " + i);
        producedCount++;
        notify();
    }

    synchronized int consume() {
        if (!produced) {
            try {
                wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        System.out.println("Consumed " + i);
        produced = false;
        notify();
        return i;
    }
}

class Producer extends Thread {
    Q q;
    int produceLimit;
```



```

    Producer(Q q, int produceLimit) {
        this.q = q;
        this.produceLimit = produceLimit;
    }

    public void run() {
        for (int i = 0; i < produceLimit; i++) {
            q.produce(i);
        }
    }
}

class Consumer extends Thread {
    Q q;
    int consumeLimit;

    Consumer(Q q, int consumeLimit) {
        this.q = q;
        this.consumeLimit = consumeLimit;
    }

    public void run() {
        for (int i = 0; i < consumeLimit; i++) {
            q.consume();
        }
    }
}

public class ProducerConsumer {
    public static void main(String args[]) {
        int maxSize = 10;
        int produceLimit = 10;
        int consumeLimit = 10;

        Q q = new Q(maxSize);
        Producer p = new Producer(q, produceLimit);
        Consumer c = new Consumer(q, consumeLimit);

        p.start();
        c.start();
    }
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac ProducerConsumer.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java ProducerConsumer
```

```

Produced 0
Consumed 0
Produced 1
Consumed 1
Produced 2
Consumed 2
Produced 3
Consumed 3
Produced 4
Consumed 4
Produced 5
Consumed 5
Produced 6
Consumed 6
Produced 7
Consumed 7
Produced 8
Consumed 8
Produced 9
Consumed 9

```

## SECTION 10

**1. Write a program to display the contents of a directory by displaying the subdirectory's name first, then the file names.**

```
import java.io.File;

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

        String directoryPath = "C:\\Users\\athul\\OneDrive\\Desktop\\java09";

        File directory = new File(directoryPath);

        if (directory.exists() && directory.isDirectory()) {
            displayContents(directory);
        } else {
            System.out.println("Invalid directory path.");
        }
    }

    public static void displayContents(File directory) {

        File[] files = directory.listFiles();

        for (File file : files) {
            if (file.isDirectory()) {
                System.out.println("Subdirectory: " + file.getName());
            }
        }

        for (File file : files) {
            if (file.isFile()) {
                System.out.println("File: " + file.getName());
            }
        }
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac DirectoryContents.java
C:\Users\athul\OneDrive\Desktop\java09>java DirectoryContents
Subdirectory: javaprograms
Subdirectory: packageprgm
File: abc.class
File: abc.java
File: AreaTriangle.class
File: AreaTriangle.java
File: arrayExcep.java
File: ArrayExceptionHandler.class
File: ArrayExceptionHandler.java
File: Circle.class
File: CircleArea.class
File: CircleArea.java
File: cloneDemo.class
File: cloneDemo.java
File: Complex.class
File: Complex.java
File: ComplexOperations.class
File: ComplexOperations.java
```

**2. Write a program to display the contents of a directory including its subdirectory content using recursive function.**

```
import java.io.File;

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

        String directoryPath = "C:/Users/athul/OneDrive/Desktop/java09";

        File directory = new File(directoryPath);

        if (directory.exists() && directory.isDirectory()) {
            displayContentsRecursive(directory, 0);
        } else {
            System.out.println("Invalid directory path.");
        }
    }

    public static void displayContentsRecursive(File directory, int depth) {

        File[] files = directory.listFiles();
        if (files != null) {
            for (File file : files) {
                if (file.isDirectory()) {

                    for (int i = 0; i < depth; i++) {
                        System.out.print("\t");
                    }
                    System.out.println("Subdirectory: " + file.getName());

                    displayContentsRecursive(file, depth + 1);
                }
            }

            for (File file : files) {
                if (file.isFile()) {

                    for (int i = 0; i < depth; i++) {
                        System.out.print("\t");
                    }
                    System.out.println("File: " + file.getName());
                }
            }
        }
    }
}
```

```

C:\Users\athul\OneDrive\Desktop\java09>javac DirectoryContentsRecursive.java

C:\Users\athul\OneDrive\Desktop\java09>java DirectoryContentsRecursive
Subdirectory: javaprograms
    Subdirectory: javademo
Subdirectory: packageprgrm
    Subdirectory: mathutils
        File: CombinationCalculator.class
        File: CombinationCalculator.java
        File: PrimeFactorsCalculator.class
        File: PrimeFactorsCalculator.java
    File: CombinationApp.class
    File: CombinationApp.java
    File: PrimeFactorsApp.class
    File: PrimeFactorsApp.java
File: abc.class
File: abc.java
File: AreaTriangle.class
File: AreaTriangle.java
File: arrayExcep.java
File: ArrayExceptionHandling.class
File: ArrayExceptionHandling.java

```

### 3. Write a program to search for a given file name in a directory (including its subdirectory) content.

```

import java.io.File;

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

        String directoryPath = "C:/Users/athul/OneDrive/Desktop/java09";
        String fileNameToSearch = "Hello.java";

        searchFile(new File(directoryPath), fileNameToSearch);
    }

    public static void searchFile(File directory, String fileNameToSearch) {
        File[] files = directory.listFiles();
        if (files != null) {
            for (File file : files) {
                if (file.isDirectory()) {
                    searchFile(file, fileNameToSearch);
                } else {
                    if (file.getName().equals(fileNameToSearch)) {
                        System.out.println("File found: " + file.getAbsolutePath());
                    }
                }
            }
        }
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac FileSearch.java

C:\Users\athul\OneDrive\Desktop\java09>java FileSearch
File found: C:\Users\athul\OneDrive\Desktop\java09\Hello.java

```

## SECTION 11

**1. Write a menu driven program to demonstrate Random Access File handling, with options for creating, deleting, writing, appending and reading the file.**

```
import java.io.File;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.util.Scanner;

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

        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Create a file");
            System.out.println("2. Delete the file");
            System.out.println("3. Write to the file");
            System.out.println("4. Append to the file");
            System.out.println("5. Read from the file");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1:
                    createFile(scanner);
                    break;
                case 2:
                    deleteFile(scanner);
                    break;
                case 3:
                    writeToFile(scanner);
                    break;
                case 4:
                    appendToFile(scanner);
                    break;
                case 5:
                    readFromFile(scanner);
                    break;
                case 6:
                    scanner.close();
                    System.out.println("Exiting program.");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice. Please enter a number between 1 and 6.");
            }
        }
    }

    private static void createFile(Scanner scanner) {
        System.out.print("Enter the file name: ");
        String fileName = scanner.nextLine();
        try {
            File file = new File(fileName);
            if (file.createNewFile()) {
                System.out.println("File created: " + file.getAbsolutePath());
            }
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

```

        } else {
            System.out.println("File already exists.");
        }
    } catch (IOException e) {
        System.out.println("An error occurred while creating the file: " + e.getMessage());
    }
}

private static void deleteFile(Scanner scanner) {
    System.out.print("Enter the file name: ");
    String fileName = scanner.nextLine();
    File file = new File(fileName);
    if (file.delete()) {
        System.out.println("File deleted: " + file.getAbsolutePath());
    } else {
        System.out.println("Failed to delete the file or the file does not exist.");
    }
}

private static void writeToFile(Scanner scanner) {
    System.out.print("Enter the file name: ");
    String fileName = scanner.nextLine();
    RandomAccessFile raf = null;
    try {
        raf = new RandomAccessFile(fileName, "rw");
        System.out.print("Enter the content to write: ");
        String content = scanner.nextLine();
        raf.writeBytes(content);
        System.out.println("Content written to the file.");
    } catch (IOException e) {
        System.out.println("An error occurred while writing to the file: " + e.getMessage());
    } finally {
        if (raf != null) {
            try {
                raf.close();
            } catch (IOException e) {
                System.out.println("An error occurred while closing the file: " + e.getMessage());
            }
        }
    }
}

private static void appendToFile(Scanner scanner) {
    System.out.print("Enter the file name: ");
    String fileName = scanner.nextLine();
    RandomAccessFile raf = null;
    try {
        raf = new RandomAccessFile(fileName, "rw");
        System.out.print("Enter the content to append: ");
        String content = scanner.nextLine();
        raf.seek(raf.length());
        raf.writeBytes(content);
        System.out.println("Content appended to the file.");
    } catch (IOException e) {
        System.out.println("An error occurred while appending to the file: " + e.getMessage());
    } finally {
        if (raf != null) {
            try {
                raf.close();
            } catch (IOException e) {
                System.out.println("An error occurred while closing the file: " + e.getMessage());
            }
        }
    }
}

```

```

    }
    }
}

private static void readFromFile(Scanner scanner) {
    System.out.print("Enter the file name: ");
    String fileName = scanner.nextLine();
    RandomAccessFile raf = null;
    try {
        raf = new RandomAccessFile(fileName, "r");
        System.out.println("Content of the file:");
        String line;
        while ((line = raf.readLine()) != null) {
            System.out.println(line);
        }
    } catch (IOException e) {
        System.out.println("An error occurred while reading from the file: " + e.getMessage());
    } finally {
        if (raf != null) {
            try {
                raf.close();
            } catch (IOException e) {
                System.out.println("An error occurred while closing the file: " + e.getMessage());
            }
        }
    }
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac RandomAccessFileDemo.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java RandomAccessFileDemo
```

```
Menu:
```

1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit

```
Enter your choice: 1
```

```
Enter the file name: java01
```

```
File already exists.
```

```
Menu:
```

1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit

```
Enter your choice: 3
```

```
Enter the file name: java01
```

```
Enter the content to write: Hello
```

```
Content written to the file.
```

```
Menu:
```

1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit

```
Enter your choice: 4
```

```
Enter the file name: java01
```

```
Enter the content to append: ....Java
```

```
Content appended to the file.
```

```

Menu:
1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit
Enter your choice: 4
Enter the file name: java01
Enter the content to append: ....Java
Content appended to the file.

Menu:
1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit
Enter your choice: 5
Enter the file name: java01
Content of the file:
Hello....Java

Menu:
1. Create a file
2. Delete the file
3. Write to the file
4. Append to the file
5. Read from the file
6. Exit
Enter your choice: 2
Enter the file name: java01
File deleted: C:\Users\athul\OneDrive\Desktop\java09\java01

```

**2. Write a program to implement a Generic method, which can display the elements of various arrays of different data types, and find the length of each array.**

```

public class GenericArrayDemo {
    public static <T> void displayAndFindLength(T[] array) {
        System.out.println("Elements of the array:");
        for (T element : array) {
            System.out.print(element + " ");
        }
        System.out.println("\nLength of the array: " + array.length);
    }

    public static void main(String[] args) {

        Integer[] intArray = {1, 2, 3, 4, 5};
        Double[] doubleArray = {1.1, 2.2, 3.3, 4.4, 5.5};
        Character[] charArray = {'a', 'b', 'c', 'd', 'e'};
        String[] stringArray = {"apple", "banana", "orange", "grape", "kiwi"};

        System.out.println("Integer Array:");
        displayAndFindLength(intArray);

        System.out.println("\nDouble Array:");
        displayAndFindLength(doubleArray);

        System.out.println("\nCharacter Array:");
        displayAndFindLength(charArray);

        System.out.println("\nString Array:");
    }
}

```



```

        displayAndFindLength(stringArray);
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac GenericArrayDemo.java

C:\Users\athul\OneDrive\Desktop\java09>java GenericArrayDemo
Integer Array:
Elements of the array:
1 2 3 4 5
Length of the array: 5

Double Array:
Elements of the array:
1.1 2.2 3.3 4.4 5.5
Length of the array: 5

Character Array:
Elements of the array:
a b c d e
Length of the array: 5

String Array:
Elements of the array:
apple banana orange grape kiwi
Length of the array: 5

```

**3. Write a program to implement a Generic class, and display the types of various parameters passed.**

```

public class GenericClassDemo<T> {
    public void displayType(T parameter) {
        System.out.println("Type of the parameter: " + parameter.getClass().getName());
    }

    public static void main(String[] args) {
        GenericClassDemo<Integer> integerInstance = new GenericClassDemo<Integer>();
        GenericClassDemo<Double> doubleInstance = new GenericClassDemo<Double>();
        GenericClassDemo<String> stringInstance = new GenericClassDemo<String>();
        GenericClassDemo<Boolean> booleanInstance = new GenericClassDemo<Boolean>();

        integerInstance.displayType(10);
        doubleInstance.displayType(3.14);
        stringInstance.displayType("Hello");
        booleanInstance.displayType(true);
    }
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac GenericClassDemo.java

C:\Users\athul\OneDrive\Desktop\java09>java GenericClassDemo
Type of the parameter: java.lang.Integer
Type of the parameter: java.lang.Double
Type of the parameter: java.lang.String
Type of the parameter: java.lang.Boolean

```

**4. Using bounded type parameters, compare the averages of various arrays.**

```

public class ArrayAverageCompare {
    public static <T extends Number> double computeAverage(T[] array) {
        double sum = 0;
    }
}

```

```

        for (T element : array) {
            sum += element.doubleValue();
        }
        return sum / array.length;
    }
}

public static <T extends Number> void compareAverages(T[] array1, T[] array2) {
    double average1 = computeAverage(array1);
    double average2 = computeAverage(array2);
    System.out.println("Average of array1: " + average1);
    System.out.println("Average of array2: " + average2);
    if (average1 > average2) {
        System.out.println("Average of array1 is greater than average of array2.");
    } else if (average1 < average2) {
        System.out.println("Average of array1 is less than average of array2.");
    } else {
        System.out.println("Average of array1 is equal to average of array2.");
    }
}

public static void main(String[] args) {
    Integer[] intArray1 = {10, 20, 30, 40, 50};
    Integer[] intArray2 = {15, 25, 35, 45, 55};
    Double[] doubleArray1 = {1.5, 2.5, 3.5, 4.5, 5.5};
    Double[] doubleArray2 = {2.0, 3.0, 4.0, 5.0, 6.0};
    System.out.println("Comparing averages of integer arrays:");
    compareAverages(intArray1, intArray2);
    System.out.println("\nComparing averages of double arrays:");
    compareAverages(doubleArray1, doubleArray2);
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac ArrayAverageCompare.java

C:\Users\athul\OneDrive\Desktop\java09>java ArrayAverageCompare
Comparing averages of integer arrays:
Average of array1: 30.0
Average of array2: 35.0
Average of array1 is less than average of array2.

Comparing averages of double arrays:
Average of array1: 3.5
Average of array2: 4.0
Average of array1 is less than average of array2.

```

## SECTION 12

### 1. Write a program to implement Serialization and De-Serialization, for an object of Student Class.

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

class Student implements Serializable {
    private static final long serialVersionUID = 1L;
    private String name;
    private int id;

    public Student(String name, int id) {
        this.name = name;
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public int getId() {
        return id;
    }

    public String toString() {
        return "Student [name=" + name + ", id=" + id + "]";
    }
}

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

        // Getting student details from user
        System.out.println("Enter student name:");
        String name = scanner.nextLine();

        System.out.println("Enter student ID:");
        int id = scanner.nextInt();

        Student student = new Student(name, id);

        // Serialization
        try {
            FileOutputStream fileOut = new FileOutputStream("student.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(student);
            out.close();
            fileOut.close();
            System.out.println("Serialized data is saved in student.ser");
        } catch (IOException i) {
            i.printStackTrace();
        }

        // Deserialization
        Student deserializedStudent = null;
        try {
            FileInputStream fileIn = new FileInputStream("student.ser");
            ObjectInputStream in = new ObjectInputStream(fileIn);
            deserializedStudent = (Student) in.readObject();
            in.close();
        }
```

```

        fileIn.close();
    } catch (IOException i) {
        i.printStackTrace();
        return;
    } catch (ClassNotFoundException c) {
        System.out.println("Student class not found");
        c.printStackTrace();
        return;
    }

    // Displaying deserialized student
    System.out.println("Deserialized Student:");
    System.out.println(deserializedStudent);

    scanner.close();
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac SerializationDemoUserInput.java

C:\Users\athul\OneDrive\Desktop\java09>java SerializationDemoUserInput
Enter student name:
Athuu
Enter student ID:
101
Serialized data is saved in student.ser
Deserialized Student:
Student [name=Athuu, id=101]

```

## 2. Write a program to implement IS-A Serialization and De-Serialization, for a Maruti Car inherited from Vehicle.

```

import java.io.*;
import java.util.Scanner;

// Base class (parent)
class Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private String make;
    private String model;

    public Vehicle(String make, String model) {
        this.make = make;
        this.model = model;
    }

    public String getMake() {
        return make;
    }

    public String getModel() {
        return model;
    }

    @Override
    public String toString() {
        return "Vehicle [make=" + make + ", model=" + model + "]";
    }
}

// Derived class (child)

```

```

class MarutiCar extends Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private int year;

    public MarutiCar(String make, String model, int year) {
        super(make, model);
        this.year = year;
    }

    public int getYear() {
        return year;
    }

    @Override
    public String toString() {
        return "MarutiCar [make=" + getMake() + ", model=" + getModel() + ", year=" + year + "]";
    }
}

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

        // Getting Maruti car details from user
        System.out.println("Enter Maruti car make:");
        String make = scanner.nextLine();

        System.out.println("Enter Maruti car model:");
        String model = scanner.nextLine();

        System.out.println("Enter Maruti car year:");
        int year = scanner.nextInt();

        MarutiCar car = new MarutiCar(make, model, year);

        // Serialization
        try {
            FileOutputStream fileOut = new FileOutputStream("maruti_car.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(car);
            out.close();
            fileOut.close();
            System.out.println("Serialized data is saved in maruti_car.ser");
        } catch (IOException i) {
            i.printStackTrace();
        }

        // Deserialization
        MarutiCar deserializedCar = null;
        try {
            FileInputStream fileIn = new FileInputStream("maruti_car.ser");
            ObjectInputStream in = new ObjectInputStream(fileIn);
            deserializedCar = (MarutiCar) in.readObject();
            in.close();
            fileIn.close();
        } catch (IOException i) {
            i.printStackTrace();
            return;
        } catch (ClassNotFoundException c) {
            System.out.println("MarutiCar class not found");
        }
    }
}

```

```

        c.printStackTrace();
        return;
    }

    // Displaying deserialized car
    System.out.println("Deserialized MarutiCar:");
    System.out.println(deserializedCar);

    scanner.close();
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac SerializationDemo.java

C:\Users\athul\OneDrive\Desktop\java09>java SerializationDemo
Enter Maruti car make:
Maruti
Enter Maruti car model:
brezza
Enter Maruti car year:
2020
Serialized data is saved in maruti_car.ser
Deserialized MarutiCar:
MarutiCar [make=Maruti, model=brezza, year=2020]

```

### 3. Write a program to implement HAS-A Serialization and De-Serialization for the Engine of a Vehicle.

```

import java.io.*;
import java.util.Scanner;
class Engine implements Serializable {
    private static final long serialVersionUID = 1L;
    private String type;
    private double horsepower;
    public Engine(String type, double horsepower) {
        this.type = type;
        this.horsepower = horsepower;
    }

    public String getType() {
        return type;
    }

    public double getHorsepower() {
        return horsepower;
    }

    public String toString() {
        return "Engine [type=" + type + ", horsepower=" + horsepower + "]";
    }
}

class Vehicle implements Serializable {
    private static final long serialVersionUID = 1L;
    private String make;
    private String model;
    private transient Engine engine; // transient keyword to prevent engine from being serialized
    public Vehicle(String make, String model, Engine engine) {
        this.make = make;
        this.model = model;
        this.engine = engine;
    }
}

```

```

public String getMake() {
    return make;
}
public String getModel() {
    return model;
}

public Engine getEngine() {
    return engine;
}
public String toString() {
    return "Vehicle [make=" + make + ", model=" + model + ", engine=" + engine + "]";
}
}

public class SerializationDemoUserInput2 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Getting engine details from user
        System.out.println("Enter engine type:");
        String type = scanner.nextLine();

        System.out.println("Enter engine horsepower:");
        double horsepower = scanner.nextDouble();
        scanner.nextLine(); // Consume newline character

        Engine engine = new Engine(type, horsepower);

        // Getting vehicle details from user
        System.out.println("Enter vehicle make:");
        String make = scanner.nextLine();

        System.out.println("Enter vehicle model:");
        String model = scanner.nextLine();

        Vehicle vehicle = new Vehicle(make, model, engine);

        // Serialization
        try {
            FileOutputStream fileOut = new FileOutputStream("vehicle.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);
            out.writeObject(vehicle);
            out.close();
            fileOut.close();
            System.out.println("Serialized data is saved in vehicle.ser");
        } catch (IOException i) {
            i.printStackTrace();
        }

        // Deserialization
        Vehicle deserializedVehicle = null;
        try {
            FileInputStream fileIn = new FileInputStream("vehicle.ser");
            ObjectInputStream in = new ObjectInputStream(fileIn);
            deserializedVehicle = (Vehicle) in.readObject();
            in.close();
            fileIn.close();
        } catch (IOException i) {
            i.printStackTrace();
            return;
        }
    }
}

```

```

    } catch (ClassNotFoundException c) {
        System.out.println("Vehicle class not found");
        c.printStackTrace();
        return;
    }

    // Displaying deserialized vehicle
    System.out.println("Deserialized Vehicle:");
    System.out.println(deserializedVehicle);

    scanner.close();
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac  SerializationDemoUserInput2.java

C:\Users\athul\OneDrive\Desktop\java09>java  SerializationDemoUserInput2
Enter engine type:
internal combustion engine
Enter engine horsepower:
80
Enter vehicle make:
honda
Enter vehicle model:
honda amaze
Serialized data is saved in vehicle.ser
Deserialized Vehicle:
Vehicle [make=honda, model=honda amaze, engine=null]

```

#### 4. Write a program to Serialize/De-Serialize selected attributes of an Employee.

```

import java.io.*;
import java.util.Scanner;

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;

    private String name;
    private int age;
    private double salary;
    private transient String address;

    public Employee(String name, int age, double salary, String address) {
        this.name = name;
        this.age = age;
        this.salary = salary;
        this.address = address;
    }

    public String getName() {
        return name;
    }

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

    public int getAge() {
        return age;
    }
}

```



```

public void setAge(int age) {
    this.age = age;
}

public double getSalary() {
    return salary;
}

public void setSalary(double salary) {
    this.salary = salary;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}
}

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

        System.out.println("Enter employee name:");
        String name = scanner.nextLine();

        System.out.println("Enter employee age:");
        int age = scanner.nextInt();

        System.out.println("Enter employee salary:");
        double salary = scanner.nextDouble();

        System.out.println("Enter employee address:");

        scanner.nextLine();
        String address = scanner.nextLine();
        Employee employee = new Employee(name, age, salary, address);
        try {
            FileOutputStream fileOut = new FileOutputStream("employee.ser");
            ObjectOutputStream out = new ObjectOutputStream(fileOut);

            out.writeObject(employee.getName());
            out.writeObject(employee.getAge());
            out.writeObject(employee.getSalary());
            out.close();
            fileOut.close();
            System.out.println("Serialized data is saved in employee.ser");
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }

    String deserializedName = null;
    int deserializedAge = 0;
    double deserializedSalary = 0.0;
    try {
        FileInputStream fileIn = new FileInputStream("employee.ser");
        ObjectInputStream in = new ObjectInputStream(fileIn);
        deserializedName = (String) in.readObject();
    }

```

```
        deserializedAge = (int) ((Integer) in.readObject());
        deserializedSalary = (double) ((Double) in.readObject());
        in.close();
        fileIn.close();
        System.out.println("Deserialized Employee:");
        System.out.println("Name: " + deserializedName);
        System.out.println("Age: " + deserializedAge);
        System.out.println("Salary: " + deserializedSalary);
    } catch (IOException ioe) {
        ioe.printStackTrace();
    } catch (ClassNotFoundException cnfe) {
        cnfe.printStackTrace();
    }
    scanner.close();
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac Main.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java Main
```

```
Enter employee name:
```

```
Aravind
```

```
Enter employee age:
```

```
28
```

```
Enter employee salary:
```

```
70000
```

```
Enter employee address:
```

```
Kannur
```

```
Serialized data is saved in employee.ser
```

```
Deserialized Employee:
```

```
Name: Aravind
```

```
Age: 28
```

```
Salary: 70000.0
```

## SECTION 13

### 1. Write a program to implement various methods of a StringBuffer class.

```
public class StringBufferExample
{
    public static void main(String[] args)
    {
        StringBuffer sb = new StringBuffer("Hello"); // Create a StringBuffer object

        sb.append(" World"); // Append method
        System.out.println("After append(): " + sb);

        sb.insert(5, " Java"); // Insert method
        System.out.println("After insert(): " + sb);

        sb.delete(5, 10); // Delete method
        System.out.println("After delete(): " + sb);

        sb.reverse(); // Reverse method
        System.out.println("After reverse(): " + sb);

        int length = sb.length(); // Length method
        System.out.println("Length of StringBuffer: " + length);

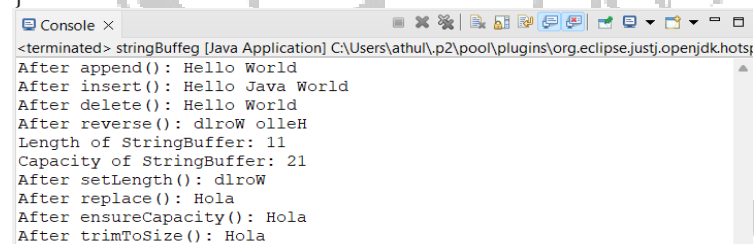
        int capacity = sb.capacity(); // Capacity method
        System.out.println("Capacity of StringBuffer: " + capacity);

        sb.setLength(5); // SetLength method
        System.out.println("After setLength(): " + sb);

        sb.replace(0, 5, "Hola"); // Replace method
        System.out.println("After replace(): " + sb);

        sb.ensureCapacity(50); // EnsureCapacity method
        System.out.println("After ensureCapacity(): " + sb);

        sb.trimToSize(); // TrimToSize method
        System.out.println("After trimToSize(): " + sb);
    }
}
```



```
<terminated> stringBuffer [Java Application] C:\Users\athul.p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full\jre\bin\java.exe
After append(): Hello World
After insert(): Hello Java World
After delete(): Hello World
After reverse(): dlroW olleH
Length of StringBuffer: 11
Capacity of StringBuffer: 21
After setLength(): dlroW
After replace(): Hola
After ensureCapacity(): Hola
After trimToSize(): Hola
```

### 2. Write a program to implement communication between a client and server client, via Socket Programming.

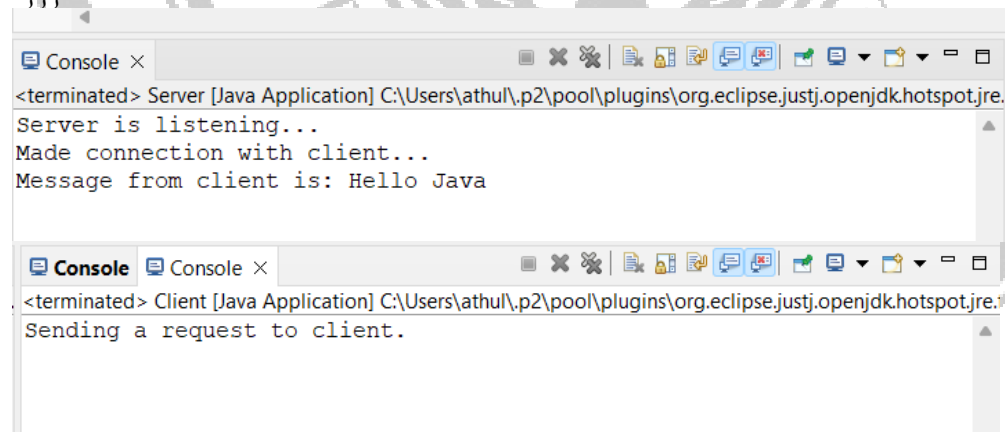
```
package publicchat;
import java.io.*;
```

```

import java.net.*;
public class Server {
    public static void main(String[] args) {
        try{
            ServerSocket ss= new ServerSocket(12345);
            System.out.println("Server is listening...");
            Socket s=ss.accept();
            System.out.println("Made connection with client...");
            ObjectInputStream dis=new ObjectInputStream(s.getInputStream());
            String str=(String)dis.readUTF();
            System.out.println("Message from client is: "+str);
            ss.close();
            dis.close();
        } catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}

package publicchat;
import java.io.*;
import java.net.*;
public class Client {
    public static void main(String[] args){
        try{
            Socket s=new Socket("localhost",12345);
            System.out.println("Sending a request to client.");
            ObjectOutputStream dout=new ObjectOutputStream(s.getOutputStream());
            dout.writeUTF("Hello Java");
            dout.close();
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}

```



### 3. Write a program to implement one-one chatting using the TCP protocol.

```

package string;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;

```

```

import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.ServerSocket;
import java.net.Socket;

public class server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(3330);
            System.out.println("Server started, waiting for clients...");

            Socket socket = serverSocket.accept();
            System.out.println("Client connected: " + socket);

            BufferedReader inputReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter outputWriter = new PrintWriter(socket.getOutputStream(), true);
            BufferedReader consoleReader = new BufferedReader(new InputStreamReader(System.in));

            String receivedMessage, sendMessage;
            while (true) {
                receivedMessage = inputReader.readLine();
                if (receivedMessage.equals("exit")) {
                    System.out.println("Client disconnected");
                    break;
                }
                System.out.println("Client: " + receivedMessage);

                System.out.print("You: ");
                sendMessage = consoleReader.readLine();
                outputWriter.println(sendMessage);
            }

            socket.close();
            serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

package string;

import java.io.BufferedReader;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintWriter;
import java.net.Socket;

public class client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 3330h);
            System.out.println("Connected to server: " + socket);

```

```

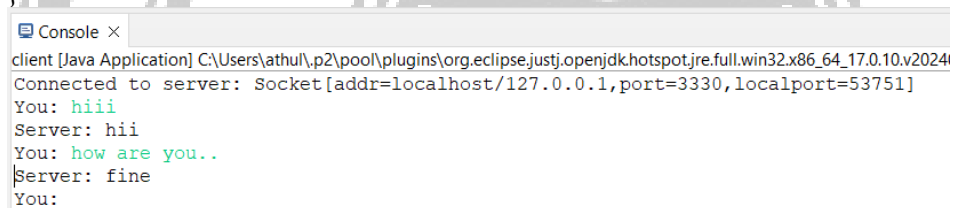
BufferedReader inputReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
PrintWriter outputWriter = new PrintWriter(socket.getOutputStream(), true);
BufferedReader consoleReader = new BufferedReader(new InputStreamReader(System.in));

String receivedMessage, sendMessage;
while (true) {
    System.out.print("You: ");
    sendMessage = consoleReader.readLine();
    outputWriter.println(sendMessage);
    if (sendMessage.equals("exit")) {
        System.out.println("Disconnected from server");
        break;
    }

    receivedMessage = inputReader.readLine();
    if (receivedMessage.equals("exit")) {
        System.out.println("Server disconnected");
        break;
    }
    System.out.println("Server: " + receivedMessage);
}

socket.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

```



```

Console ×
client [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.10.v2024
Connected to server: Socket[addr=localhost/127.0.0.1,port=3330,localport=53751]
You: hiii
Server: hii
You: how are you..
Server: fine
You:

```

**4. Write a program to accept a file name given in the client, and to download the content from the server side.**

```

// Server.java
import java.io.*;
import java.net.*;

public class Server {
    public static void main(String[] args) {
        try {
            ServerSocket serverSocket = new ServerSocket(3333);
            System.out.println("Server started, waiting for clients...");

            Socket socket = serverSocket.accept();
            System.out.println("Client connected: " + socket);

            BufferedReader inputReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter outputWriter = new PrintWriter(socket.getOutputStream(), true);

            String fileName = inputReader.readLine();
            System.out.println("Client requested file: " + fileName);

            File file = new File(fileName);
            if (file.exists()) {

```

```

        BufferedReader fileReader = new BufferedReader(new FileReader(file));
        String line;
        while ((line = fileReader.readLine()) != null) {
            outputWriter.println(line);
        }
        fileReader.close();
        System.out.println("File sent successfully");
    } else {
        outputWriter.println("File not found");
        System.out.println("File not found");
    }
}

socket.close();
serverSocket.close();
} catch (IOException e) {
    e.printStackTrace();
}
}
}

// Client.java
import java.io.*;
import java.net.*;

public class Client {
    public static void main(String[] args) {
        try {
            Socket socket = new Socket("localhost", 3333);
            System.out.println("Connected to server: " + socket);

            BufferedReader inputReader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            PrintWriter outputWriter = new PrintWriter(socket.getOutputStream(), true);
            BufferedReader consoleReader = new BufferedReader(new InputStreamReader(System.in));

            System.out.print("Enter file name to download: ");
            String fileName = consoleReader.readLine();
            outputWriter.println(fileName);

            String line;
            while ((line = inputReader.readLine()) != null) {
                if (line.equals("File not found")) {
                    System.out.println("File not found on server");
                    break;
                }
                System.out.println("Received: " + line);
            }

            socket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

```
C:\Users\athul\OneDrive\Desktop\java09>javac Server.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac Client.java
```

```
C:\Users\athul\OneDrive\Desktop\java09>java Client
Connected to server: Socket[addr=localhost/127.0.0.1,port=3333,localport=54641]
Enter file name to download: Hello.java
```

## SECTION 14

### 1. Write a program to implement public chatting.

```
import java.net.*;
import java.io.*;
import java.util.*;
public class GroupChat
{
    private static final String TERMINATE = "Exit";
    static String name;
    static volatile boolean finished = false;
    public static void main(String[] args)
    {
        if (args.length != 2)/(239.0.0.0 1234)
            System.out.println("Two arguments required: <multicast-host> <portnumber>");
        else
        {
            try
            {
                InetAddress group = InetAddress.getByAddress(args[0]);
                int port = Integer.parseInt(args[1]);
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter your name: ");
                name = sc.nextLine();
                MulticastSocket socket = new MulticastSocket(port);
                socket.setTimeToLive(0);
                socket.joinGroup(group);
                Thread t = new Thread(new ReadThread(socket,group,port));
                t.start();
                System.out.println("Start typing messages...\n");
                while(true)
                {
                    String message;
                    message = sc.nextLine();
                    if(message.equalsIgnoreCase(GroupChat.TERMINATE))

                    {
                        finished = true;
                        socket.leaveGroup(group);
                        socket.close();
                        break;
                    }
                    message = name + ": " + message;
                    byte[] buffer = message.getBytes();
                    DatagramPacket datagram = new
                    DatagramPacket(buffer,buffer.length,group,port);
                    socket.send(datagram);
                }
            }
            catch(SocketException se)
            {
                System.out.println("Error creating socket");
                se.printStackTrace();
            }
            catch(IOException ie)
            {
                System.out.println("Error reading/writing from/to socket");
                ie.printStackTrace();
            }
        }
    }
}
```



```

}
}
}
class ReadThread implements Runnable
{
private MulticastSocket socket;
private InetAddress group;
private int port;
private static final int MAX_LEN = 1000;
ReadThread(MulticastSocket socket,InetAddress group,int port)
{
this.socket = socket;
this.group = group;
this.port = port;
}

@Override
public void run()
{
while(!GroupChat.finished)

{
byte[] buffer = new byte[ReadThread.MAX_LEN];//
DatagramPacket datagram = new
DatagramPacket(buffer,buffer.length,group,port);
String message;
try
{
socket.receive(datagram);
message = new String(buffer,0,datagram.getLength(),"UTF8");//parameters?

//if(!message.startsWith(GroupChat.name))/?
System.out.println(message);
}
catch(IOException e)
{
System.out.println("Socket closed!");
}
}
}
}

```

```

C:\Users\athul\OneDrive\Desktop\java09>javac GroupChat.java
Note: GroupChat.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java GroupChat
Two arguments required: <multicast-host> <portnumber>

```

## 2. Accept a number from the client, and print its factorial in the server.

```

import java.io.*;
import java.net.*;

public class SockSer {
    public static void main(String[] args) {
        try {
            ServerSocket ss = new ServerSocket(12345);
            System.out.println("Waiting patiently for connection from client");

```

```

Socket s = ss.accept();
System.out.println("Made connection with client");

BufferedReader reader = new BufferedReader(new InputStreamReader(s.getInputStream()));
String input = reader.readLine(); // Read number from client

int number = Integer.parseInt(input);
long factorial = calculateFactorial(number);

PrintWriter writer = new PrintWriter(s.getOutputStream(), true);
writer.println(factorial);

System.out.println("Factorial sent to client: " + factorial);

reader.close();
writer.close();
s.close();
ss.close();
} catch (IOException e) {
    e.printStackTrace();
}
}

private static long calculateFactorial(int number) {
    if (number == 0 || number == 1) {
        return 1;
    }
    long factorial = 1;
    for (int i = 2; i <= number; i++) {
        factorial *= i;
    }
    return factorial;
}

import java.io.*;
import java.net.*;

public class SockCli {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("localhost", 12345);
            System.out.println("Sending a request to server");

            BufferedReader userInputReader = new BufferedReader(new InputStreamReader(System.in));
            System.out.print("Enter a number to calculate its factorial: ");
            String userInput = userInputReader.readLine();

            PrintWriter writer = new PrintWriter(s.getOutputStream(), true);
            writer.println(userInput);

            BufferedReader reader = new BufferedReader(new InputStreamReader(s.getInputStream()));
            String response = reader.readLine();

            System.out.println("Factorial received from server: " + response);

            userInputReader.close();
            writer.close();
            reader.close();
        }
    }
}

```

```
s.close();  
} catch (IOException e) {  
    e.printStackTrace();  
}  
}
```

Console ×

<terminated> FactorialServer [Java Application] C:\Users\athul\.p2\pool\plugins\org.eclipse.justj.openjdk.hot  
Waiting patiently for connection from client  
Made connection with client  
Factorial sent to client: 3628800

Console Console ×

<terminated> FactorialClient [Java Application] C:\Users\athul\.p2\pool\plugins\org.eclipse.justj.openjdk.hot  
Sending a request to server  
Enter a number to calculate its factorial: 10  
Factorial received from server: 3628800



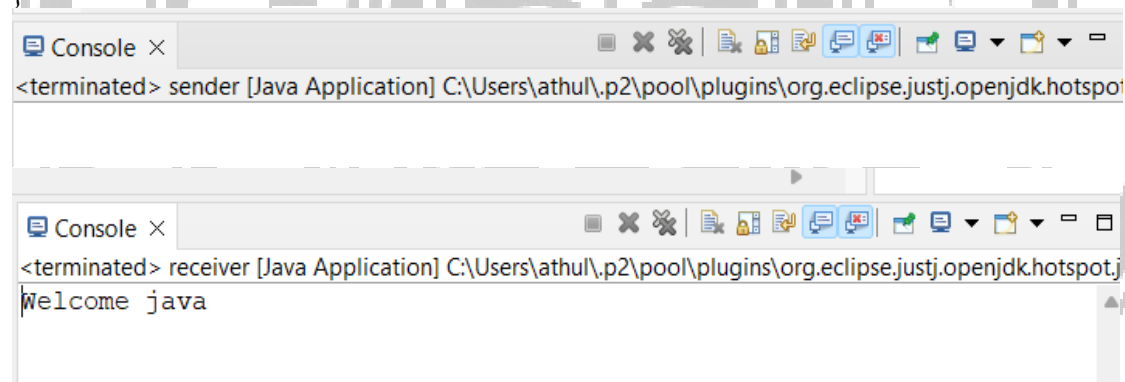
## SECTION 15

### 1. Write a program to implement connection less communication.

```
import java.net.*;
public class DSender{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket();
        String str = "Welcome java";
        InetAddress ip = InetAddress.getByName("127.0.0.1");

        DatagramPacket dp = new DatagramPacket(str.getBytes(), str.length(), ip, 3000);
        ds.send(dp);
        ds.close();
    }
}
```

```
import java.net.*;
public class DReceiver{
    public static void main(String[] args) throws Exception {
        DatagramSocket ds = new DatagramSocket(3000);
        byte[] buf = new byte[1024];
        DatagramPacket dp = new DatagramPacket(buf, 1024);
        ds.receive(dp);
        String str = new String(dp.getData(), 0, dp.getLength());
        System.out.println(str);
        ds.close();
    }
}
```



### 2. Write a program to get protocol, file name, host, path and port of a given URL.

```
import java.net.*;
public class urldemo{
    public static void main(String[] args){
        try{
            URL url=new URL("http://www.javatpoint.com/java-tutorial");

            System.out.println("Protocol: "+url.getProtocol());
            System.out.println("Host Name: "+url.getHost());
            System.out.println("Port Number: "+url.getPort());
            System.out.println("File Name: "+url.getFile());

        }catch(Exception e){System.out.println(e);}
    }
}
```

```
Console ×
<terminated> urldemo [Java Application] C:\Users\athul\.p2\pool\plugins\org.eclipse.justj.openjdk.hotsp
Protocol: http
Host Name: www.javatpoint.com
Port Number: -1
File Name: /java-tutorial
```

### 3. Write a program to download a file from a given URL.

```
import java.io.FileOutputStream;
import java.io.InputStream;
import java.net.URL;
import java.net.URLConnection;

public class FileDownloader {

    public static void main(String[] args) {
        String fileURL = "https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy.pdf";
        String saveDir = "C:\\Users\\athul\\Downloads\\";
        try {
            downloadFile(fileURL, saveDir);
            System.out.println("File downloaded successfully!");
        } catch (Exception e) {
            System.err.println("Error downloading file: " + e.getMessage());
            e.printStackTrace();
        }
    }

    public static void downloadFile(String fileURL, String saveDir) throws Exception {
        URL url = new URL(fileURL);
        URLConnection conn = url.openConnection();
        InputStream inputStream = conn.getInputStream();

        if (!saveDir.endsWith("\\")) {
            saveDir += "\\";
        }
        String fileName = fileURL.substring(fileURL.lastIndexOf("/") + 1);
        String saveFilePath = saveDir + fileName;

        try (FileOutputStream outputStream = new FileOutputStream(saveFilePath)) {
            int bytesRead;
            byte[] buffer = new byte[1024];
            while ((bytesRead = inputStream.read(buffer)) != -1) {
                outputStream.write(buffer, 0, bytesRead);
            }
        }

        inputStream.close();
    }
}
```

```
C:\Users\athul\OneDrive\Desktop\java09>javac FileDownloader.java
Note: FileDownloader.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\athul\OneDrive\Desktop\java09>java FileDownloader
File downloaded successfully!
```

#### 4. Implement Two- way Communication using UDP Protocol.

```
package udp;

import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;

public class sevrer {
    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            socket = new DatagramSocket(9872);
            System.out.println("Server started...");

            while (true) {
                byte[] receiveData = new byte[1024];
                DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
                socket.receive(receivePacket);

                String receivedMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());
                System.out.println("Client: " + receivedMessage);
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (socket != null) {
                socket.close();
            }
        }
    }
}

package udp;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class client {
    public static void main(String[] args) {
        DatagramSocket socket = null;

        try {
            socket = new DatagramSocket();
            BufferedReader userInput = new BufferedReader(new InputStreamReader(System.in));
            while (true) {

                System.out.print("You: ");
                String message = userInput.readLine();
                byte[] sendData = message.getBytes();
                InetAddress serverAddress = InetAddress.getByName("localhost");
                int serverPort = 9872;
```

```
        DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,  
serverAddress, serverPort);  
        socket.send(sendPacket);  
    }  
    } catch (IOException e) {  
        e.printStackTrace();  
    } finally {  
        if (socket != null) {  
            socket.close();  
        }  
    }  
}  
}}
```

Console ×

sevrer [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.  
Server started...  
Client: hii  
  
Client: hello  
  
Client: hello world..



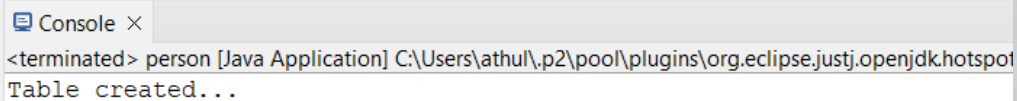
## SECTION 16

1. Write a program to create a table Person ( P\_Id(Primary), P\_Name, P\_age, P\_address, P\_DOB), insert records, and display the records.

```
package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class person {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("create table person(pid varchar2(20) primary key,pname varchar2(20),page
number,paddress varchar2(20),pdob number)");
            System.out.println("Table created...");
            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}
```



```
package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class addperson {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL",
"hr", "1234rh#");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("insert into person values('P01','Athu',22,'Kannur',2001)");
            System.out.println("Value inserted...");
            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}
```



```
Console ×
<terminated> addperson [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj.openjdk
```

```
Value inserted...
```

```
SQL> select * from person;
```

PID	PNAME	PAGE	ADDRESS
P00B			
P01	Athu	22	Kannur
2001			

2. Assume that Login is a table which has Uname, Upass. Check whether a record with “Uname=”Alexa” and “UPass=”Siri123#”is present in the table.

```
package jdbc;
import java.sql.*;
```

```
public class loginalexa {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");

            String username = "alexa";
            String password = "siri123#";
            String query = "SELECT COUNT(*) FROM login WHERE uname = '" + username + "' AND upass = '"
+ password + "'";
            Statement stmt = con.createStatement();

            ResultSet rs = stmt.executeQuery(query);

            rs.next();
            int count = rs.getInt(1);

            if (count > 0) {
                System.out.println("Record found for username 'alexa' and password 'siri123#'");
            } else {
                System.out.println("No record found for username 'alexa' and password 'siri123#'");
            }
        }

        rs.close();
        stmt.close();
        con.close();
    } catch (ClassNotFoundException | SQLException e) {
        System.out.println(e);
    }
}
}
```

```
Console ×
<terminated> loginalexa [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj.openjdk.hot
Record found for username 'alexa' and password 'siri123#'
```

## SECTION 17

### 1. Construct the following tables:

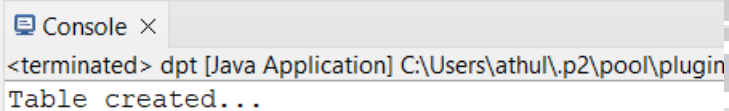
**Department (dno(Primary), dname, dloc)**

**Emp ( eno(Primary), ename, esal ,dno(Foreign))**

```
package jdbc;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;

public class dpt {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","hr","1234rh#");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("create table dpt(dno varchar2(20) primary key,dname varchar2(20),dloc varchar2(20))");
            System.out.println("Table created...");
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

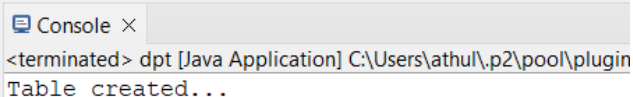


Console ×  
<terminated> dpt [Java Application] C:\Users\athul\.p2\pool\plugin  
Table created...

```
package jdbc;

import java.sql.Statement;
import java.sql.Connection;
import java.sql.DriverManager;

public class empl {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con =
            DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL","hr","1234rh#");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("create table empl(emp varchar2(20) primary key,ename varchar2(20),esal
            varchar2(20),dno varchar2(20) references Deptmnt(dno))");
            System.out.println("Table created...");
            con.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



Console ×  
<terminated> dpt [Java Application] C:\Users\athul\.p2\pool\plugin  
Table created...

### 2. Write a program for displaying information in the following order from the above tables:

eno	ename	esal	dname	dloc
101	Chetan	10,000	Civil	Kochi
102	Amish	20,000	Accounts	Delhi

```
package jdbc;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;
public class dptemp {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT e.eno, e.ename, e.esal, d.dname, d.dloc FROM empl e, dpt
d WHERE e.dno = d.dno");

            ResultSetMetaData md = rs.getMetaData();

            for (int i = 1; i <= md.getColumnCount(); i++) {
                System.out.print(" " + md.getColumnName(i));
            }
            System.out.println("\n");
            System.out.println("-----");

            int rowCount = 0;
            while (rs.next()) {
                rowCount++;
                for (int i = 1; i <= md.getColumnCount(); i++) {
                    System.out.print(rs.getString(i) + "\t");
                }
                System.out.println("\n");
            }

            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}
```

Console ×

<terminated> dptemp [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj

ENO ENAME ESAL DNAME DLOC

-----

101	chetan	10000	civil	kochi
102	amish	20000	accounts	delhi

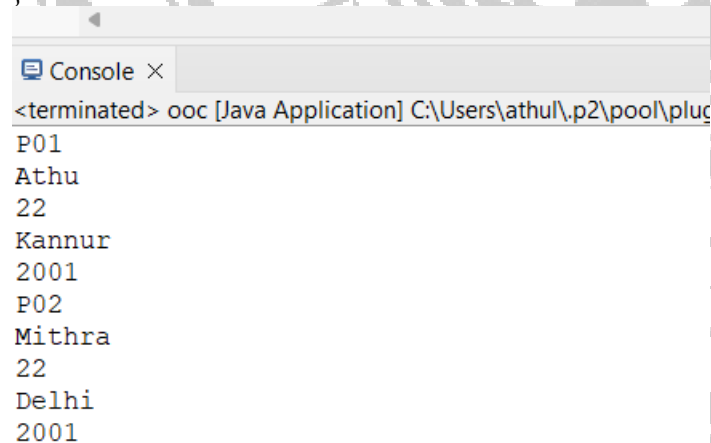
## SECTION 18

### 1. Program to implement database connectivity using object oriented concepts.

```
package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.Statement;

public class ooc {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("select * from person");
            while(rs.next()) {
                System.out.println(rs.getString(1));
                System.out.println(rs.getString(2));
                System.out.println(rs.getInt(3));
                System.out.println(rs.getString(4));
                System.out.println(rs.getInt(5));
            }
            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}
```



```
<terminated> ooc [Java Application] C:\Users\athul\p2\pool\plug
P01
Athu
22
Kannur
2001
P02
Mithra
22
Delhi
2001
```

## SECTION 19

**1. Write a JDBC program with Parametrized queries to update a given record (Rani's salary to 15,000) in the Emp table.**

```
package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class updatesal {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            Statement stmt = con.createStatement();
            PreparedStatement pst=con.prepareStatement("update empl set esal=? where eno=?");
            pst.setInt(1,15000);//1 specifies the first parameter in the query i.e. name
            pst.setString(2,"103");

            int i=pst.executeUpdate();
            System.out.println(i+" records updated");

            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}
```

Console ×  
<terminated> updatesal [Java Application] C:\Users\athul\.p2\pool\plugins\o  
1 records updated

ENO	ENAME	ESAL	DNO
101	chetan	10000	d01
102	amish	20000	d02
103	rani	15000	d02

**2. Write a JDBC program with Parametrized queries to list the records of Emp table which has records whose names start with the alphabet "R".**

```
package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

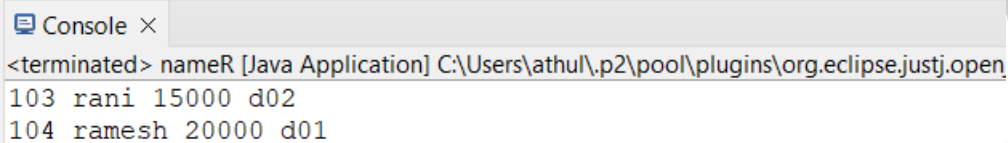
public class nameR {
```

```

public static void main(String args[]) {
    try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
        Statement stmt = con.createStatement();
        PreparedStatement pst = con.prepareStatement("select * from empl where ename like 'r%'");
        ResultSet rs = pst.executeQuery();
        while (rs.next()) {
            System.out.println(rs.getString(1) + " " + rs.getString(2) + " " + rs.getInt(3) + " " + rs.getString(4));
        }

        con.close();
    } catch (ClassNotFoundException | SQLException e) {
        System.out.println(e);
    }
}

```



```

<terminated> nameR [Java Application] C:\Users\athul\.p2\pool\plugins\org.eclipse.justj.open
103 rani 15000 d02
104 ramesh 20000 d01

```

**3. Write a JDBC program with PreparedStatement to delete the records of Emp table which has records whose salary is less than 10,000.**

```

package jdbc;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class delemp1 {
    public static void main(String args[]) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            Statement stmt = con.createStatement();
            PreparedStatement pst=con.prepareStatement("delete from empl where esal<?");
            pst.setInt(1,10000);

            int i=pst.executeUpdate();
            System.out.println(i+" records deleted");

            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println(e);
        }
    }
}

```

```
SQL> select * from empl where esal<10000;
```

ENO	ENAME	ESAL	DNO
105	Anu	8000	d01
106	Manu	7000	d02

Console ×

```
<terminated> delemp1 [Java Application] C:\Users\athul\p2\pool\plugins\org.e  
2 records deleted
```



## SECTION 20

### 1. Implement a JDBC program which uses a Stored Procedure to insert records into Department table.

```
create or replace procedure "insertR"
(did IN VARCHAR2,
dname IN VARCHAR2,
dloc IN VARCHAR2)
is
begin
insert into dpt values(did,dname,dloc);
end;
/

package jdbc;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;

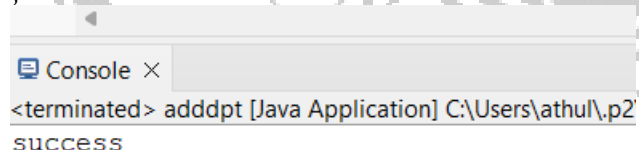
public class adddpt {
    public static void main(String[] args) throws Exception{

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL",
"hr", "1234rh#");

        CallableStatement stmt=con.prepareCall("{call insertR(?,?,?)}");
        stmt.setString(1,"d03");
        stmt.setString(2,"cs");
        stmt.setString(3,"clct");

        stmt.execute();

        System.out.println("success");
    }
}
```



Console ×  
<terminated> adddpt [Java Application] C:\Users\athul\.p2  
success

```
SQL> SELECT * FROM DPT;
```

DNO	DNAME	DLOC
d01	civil	kochi
d02	accounts	delhi
d03	cs	clct

### 2. Use Callable statement to implement a Stored Procedure to display the Ename and Salary of all employees.

PROCEDURE:

```
CREATE OR REPLACE PROCEDURE displayEmployees (cur_out OUT SYS_REFCURSOR)
IS
BEGIN
```



```

OPEN cur_out FOR
SELECT ename, esal FROM empl;
END;
/

```

CODE:

```

package jdbc;
import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;

import oracle.jdbc.OracleTypes;

public class DisplayEmployees {

    public static void main(String[] args) {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL", "hr",
"1234rh#");
            CallableStatement cstmt = con.prepareCall("{call displayEmployees(?)}")
;
            cstmt.registerOutParameter(1, OracleTypes.CURSOR);

            cstmt.execute();

            ResultSet rs = (ResultSet) cstmt.getObject(1);
            while (rs.next()) {
                String ename = rs.getString("ename");
                double salary = rs.getDouble("esal");
                System.out.println("Employee Name: " + ename + ", Salary: " + salary);
            }
            System.out.println("Stored procedure executed successfully.");
        } catch (SQLException | ClassNotFoundException e) {
            System.err.println("Error executing stored procedure: " + e.getMessage());
            e.printStackTrace();
        }
    }
}

```

```

Console ×
<terminated> DisplayEmployees [Java Application] C:\Users\athu
Employee Name: chetan, Salary: 10000.0
Employee Name: amish, Salary: 20000.0
Employee Name: rani, Salary: 15000.0
Employee Name: ramesh, Salary: 20000.0
Employee Name: Anu, Salary: 8000.0
Employee Name: Manu, Salary: 7000.0
Stored procedure executed successfully.

```

### 3. Write a JDBC program to implement Transaction Management in the Department table.

```

package jdbc;

import java.sql.Statement;
import java.sql.Connection;

```

```

import java.sql.DriverManager;

public class transactiondpt {
    public static void main(String[] args) throws Exception{

        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL",
"hr", "1234rh#");
        con.setAutoCommit(false);

        Statement stmt=con.createStatement();
        stmt.executeUpdate("insert into dpt values('d04','mlylm','ernklm')");

        con.commit();
        con.close();
    }
}

```

```
SQL> select * from dpt;
```

DNO	DNAME	DLOC
d01	civil	kochi
d02	accounts	delhi
d03	cs	clct
d04	mlylm	ernklm

#### 4. Write a JDBC program to depict the usage of SQLException Class and SQLWarning Class.

```

package jdbc;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class exception {
    public static void main(String[] args) {
        Connection connection = null;
        Statement statement = null;
        ResultSet resultSet = null;
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection connection1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:ORCL",
"hr", "1234rh#");
            Statement statement1 = connection1.createStatement();

            ResultSet resultSet1 = statement1.executeQuery("SELECT * FROM flower");

            while (resultSet1.next()) {

            }
        } catch (ClassNotFoundException e) {
            System.err.println("Failed to load Oracle JDBC driver");
            e.printStackTrace();
        } catch (SQLException e) {

```

```

// Handling SQLException
System.err.println("SQLException occurred:");
while (e != null) {
    System.err.println("SQLState: " + e.getSQLState());
    System.err.println("Error Code: " + e.getErrorCode());
    System.err.println("Message: " + e.getMessage());
    e = e.getNextException();
}
} finally {
    // Closing resources in finally block
    try {
        if (resultSet != null) {
            resultSet.close();
        }
        if (statement != null) {
            statement.close();
        }
        if (connection != null) {
            connection.close();
        }
    } catch (SQLException e) {
        // Handling SQLException while closing resources
        e.printStackTrace();
    }
}
}
}

```

Console ×

```

<terminated> exception [Java Application] C:\Users\athul\p2\pool\plugins\org.eclipse.justj.open
SQLException occurred:
SQLState: 42000
Error Code: 942
Message: ORA-00942: table or view does not exist

https://docs.oracle.com/error-help/db/ora-00942/

```