Exp. No: 1	**Basic Java Programs**
Date: 17.8.22	

Aim:

To implement basic Java programs and test them successfully.

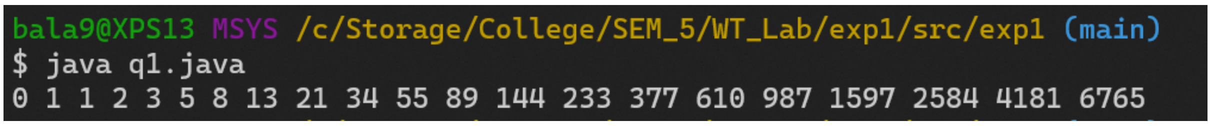
Program:

Write a java program to display Fibonacci series

Source Code:

```
public class q1 {
    public static void main(String[] args) {
        int a = 0, b = 1;
        int c;
        System.out.print(a + " " + b + " ");
        int i;
        for (i = 2; i <= 20; i++) {
            c = a + b;
            System.out.print(c + " ");
            a = b;
            b = c;
        }
    }
}
```

Output:



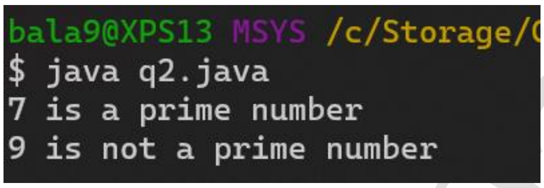
```
bala9@XPS13 MSYS /c/Storage/College/SEM_5/WT_Lab/exp1/src/exp1 (main)
$ java q1.java
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
```

2. Write a java program to check whether a number is prime or not

Source Code:

```
public class q2 {

    public static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        // check prime till square root of n
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        int x = 7;
        int y = 9;
        if (isPrime(x)) {
            System.out.println(x + " is a prime number");
        } else {
            System.out.println(x + " is not a prime number");
        }
        if (isPrime(y)) {
            System.out.println(y + " is a prime number");
        } else {
            System.out.println(y + " is not a prime number");
        }

    }
}
```

Output:



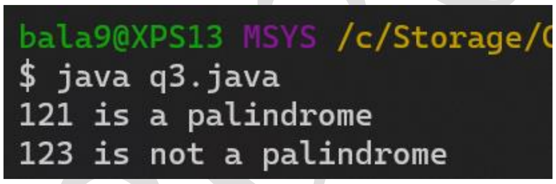
```
bala9@XPS13 MSYS /c/Storage/
$ java q2.java
7 is a prime number
9 is not a prime number
```

3. Write a java program to check whether a number is palindrome or not.

Source Code

```
public class q3 {
    public static boolean isPalindrome(int n) {
        int temp = n;
        int rev = 0;
        while (temp != 0) {
            int rem = temp % 10;
            rev = rev * 10 + rem;
            temp /= 10;
        }
        if (rev == n) {
            return true;
        } else {
            return false;
        }
    }
    public static void main(String[] args) {
        int num1 = 121;
        int num2 = 123;
        if (isPalindrome(num1)) {
            System.out.println(num1 + " is a palindrome");
        } else {
            System.out.println(num1 + " is not a palindrome");
        }
        if (isPalindrome(num2)) {
            System.out.println(num2 + " is a palindrome");
        } else {
            System.out.println(num2 + " is not a palindrome");
        }
    }
}
```

Output:



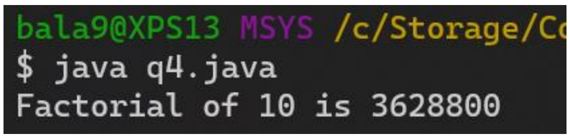
```
bala9@XPS13 MSYS /c/Storage/C
$ java q3.java
121 is a palindrome
123 is not a palindrome
```

4. Write a program to find a factorial of a number

Source Code

```
public class q4 {
    public static void main(String[] args) {
        int x = 10;
        int fact = 1;
        for (int i = 1; i <= x; i++) {
            fact *= i;
        }
        System.out.println("Factorial of " + x + " is " + fact);

    }
}
```

Output:



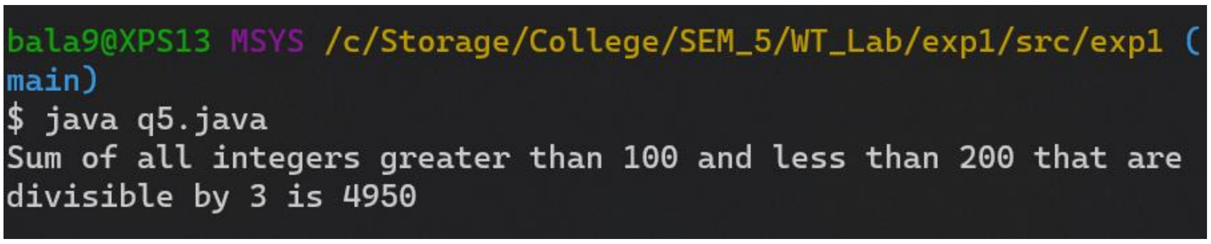
```
bala9@XPS13 MSYS /c/Storage/Co
$ java q4.java
Factorial of 10 is 3628800
```

5. Write a program to find sum of all integers greater than 100 and less than 200 that are divisible by 3

Source Code

```
public class q5 {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 101; i < 200; i++) {
            if (i % 3 == 0) {
                sum += i;
            }
        }
        System.out.println("Sum of all integers greater than 100 and less than 200 that are
divisible by 3 is " + sum);
    }
}
```

Output:



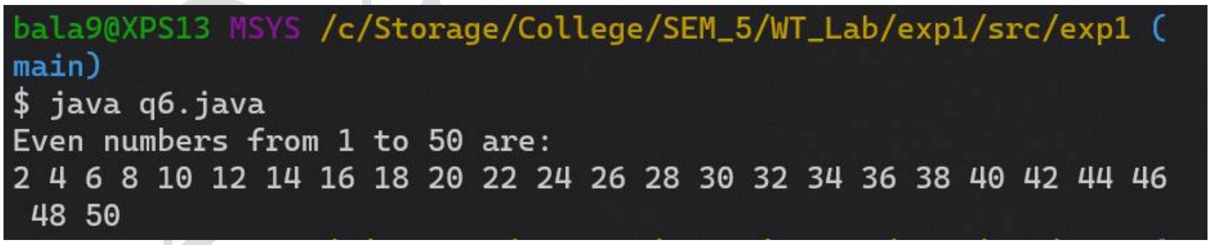
```
bala9@XPS13 MSYS /c/Storage/College/SEM_5/WT_Lab/exp1/src/exp1 (
main)
$ java q5.java
Sum of all integers greater than 100 and less than 200 that are
divisible by 3 is 4950
```

6. Write a program to print even numbers between 1 to 50

Source Code

```
public class q6 {
    public static void main(String[] args) {
        System.out.println("Even numbers from 1 to 50 are:");
        for (int i = 1; i <= 50; i++) {
            if (i % 2 == 0) {
                System.out.print(i + " ");
            }
        }
    }
}
```

Output:



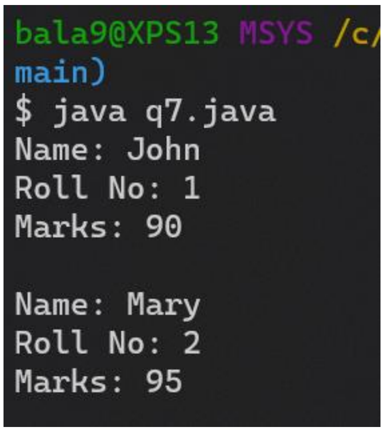
```
bala9@XPS13 MSYS /c/Storage/College/SEM_5/WT_Lab/exp1/src/exp1 (
main)
$ java q6.java
Even numbers from 1 to 50 are:
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46
 48 50
```

7. Write a program to display the student details.

Source Code

```
public class q7 {
    String name;
    int rollNo;
    int marks;
    void set(String name, int rollNo, int marks) {
        this.name = name;
        this.rollNo = rollNo;
        this.marks = marks;
    }
    void studentDetails() {
        System.out.println("Name: " + name);
        System.out.println("Roll No: " + rollNo);
        System.out.println("Marks: " + marks);
        System.out.println();
    }
    public static void main(String[] args) {
        q7 x = new q7();
        q7 y = new q7();
        x.set("John", 1, 90);
        y.set("Mary", 2, 95);
        x.studentDetails();
        y.studentDetails();

    }
}
```

<u>Output:</u>



```
bala9@XPS13 MSYS /c/
main)
$ java q7.java
Name: John
Roll No: 1
Marks: 90

Name: Mary
Roll No: 2
Marks: 95
```

8. Write a java program with a class bank account with the following methods

Credit

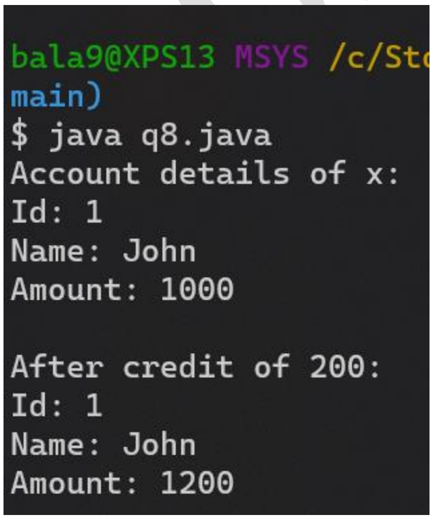
Debit

Display

<u>Source Code</u>

```
public class q8 {
    int id;
    String name;
    int amount;
    void credit(int amount) {
        this.amount += amount;
    }
    void debit(int amount) {
        if (this.amount < amount) {
            System.out.println("Amount withdrawn exceeds the current balance!");
        } else {
            this.amount -= amount;
        }
    }
    void display() {
        System.out.println("Id: " + id);
        System.out.println("Name: " + name);
        System.out.println("Amount: " + amount);
        System.out.println();
    }
    public static void main(String[] args) {
        q8 x = new q8();
        System.out.println("Account details of x:");
        x.id = 1;
        x.name = "John";
        x.amount = 1000;
        x.display();
        System.out.println("After credit of 200:");
        x.credit(200);
        x.display();
    }
}
```

Output:



```
bala9@XPS13 MSYS /c/St
main)
$ java q8.java
Account details of x:
Id: 1
Name: John
Amount: 1000

After credit of 200:
Id: 1
Name: John
Amount: 1200
```

Result:

Thus, basic java programs, has been implemented successfully.

Exp. No: 2	Basic Java Programs - II
Date: .24.8.22	

Aim:

To implement basic java programs and test them successfully.

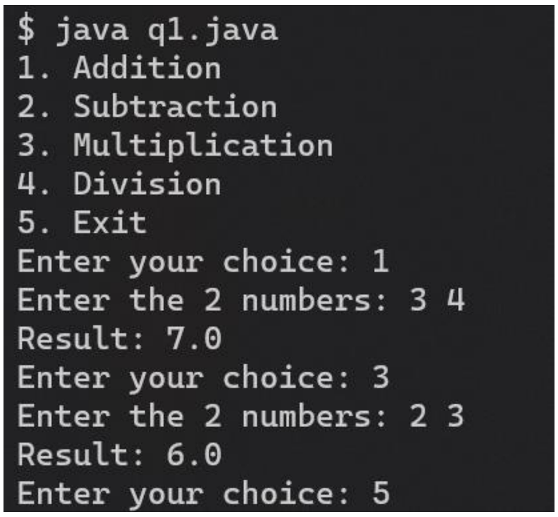
Program:

1. Write a java program to implement the calculator functionalities.

Source Code:

```
import java.util.Scanner;


public class q1 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int choice = 0;
        double num1, num2, result = 0;
        System.out.println("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5. Exit");

        while (choice != 5) {
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            System.out.print("Enter the 2 numbers: ");
            num1 = sc.nextDouble();
            num2 = sc.nextDouble();
            switch (choice) {
                case 1:
                    result = num1 + num2;
                break;
            case 2:
                result = num1 - num2;
                break;
            case 3:
                result = num1 * num2;
                break;
            case 4:
                result = num1 / num2;
                break;
            default:
                System.out.println("Invalid choice");
        }
        System.out.println("Result: " + result);

    }

    sc.close();
  }
}
```

Output:



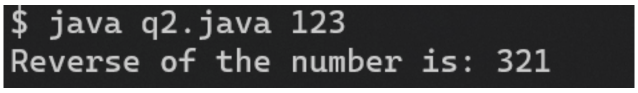
```
$ java q1.java
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Enter your choice: 1
Enter the 2 numbers: 3 4
Result: 7.0
Enter your choice: 3
Enter the 2 numbers: 2 3
Result: 6.0
Enter your choice: 5
```

2. Write a java program to reverse a number.

Source Code:

```
public class q2 {
    public static void main(String[] args) {
        int num = Integer.parseInt(args[0]);
        int rev = 0;
        while (num != 0) {
            rev = rev * 10 + num % 10;
            num /= 10;
        }
        System.out.println("Reverse of the number is: " + rev);
    }
}
```

Output:



```
$ java q2.java 123
Reverse of the number is: 321
```

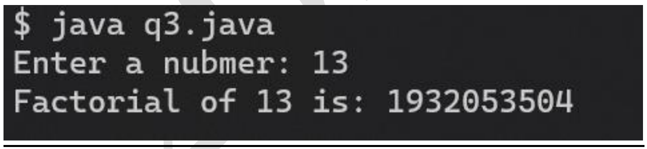
3. Write a program to find a factorial of a number in recursion.

Source Code

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

```
public class q3 {
    static int factorial(int n) {
        if (n == 0)
            return 1;
        else
            return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a nubmer: ");
        int num = sc.nextInt();
        System.out.println("Factorial of " + num + " is: " + factorial(num));
        sc.close();
    }
}
```

Output:



```
$ java q3.java
Enter a nubmer: 13
Factorial of 13 is: 1932053504
```

4. Write a java program to calculate the area of the following:

    i. Square

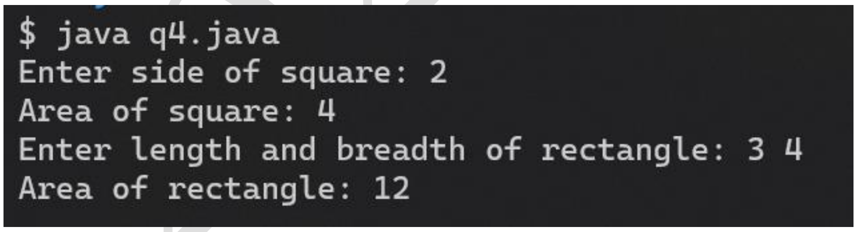
    ii. rectangle

Source Code

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

```
public class q4 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter side of square: ");
        int side = sc.nextInt();
        System.out.println("Area of square: " + side * side);

        System.out.print("Enter length and breadth of rectangle: ");
        int length = sc.nextInt();
        int breadth = sc.nextInt();
        System.out.println("Area of rectangle: " + length * breadth);

        sc.close();

    }
}
```

Output:



```
$ java q4.java
Enter side of square: 2
Area of square: 4
Enter length and breadth of rectangle: 3 4
Area of rectangle: 12
```

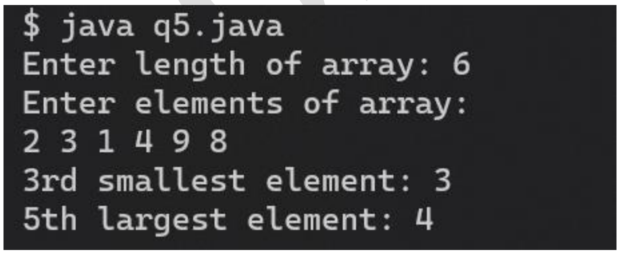
5. Write a java program to find the 5th largest and 3rd smallest element in an array.

Source Code

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

```
public class q5 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter length of array: ");
        int n = sc.nextInt();
        int arr[] = new int[n];
        System.out.println("Enter elements of array: ");
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }
        Arrays.sort(arr);
        System.out.println("3rd smallest element: " + arr[2]);
        System.out.println("5th largest element: " + arr[n - 3]);

        sc.close();
    }
}
```

Output:



```
$ java q5.java
Enter length of array: 6
Enter elements of array:
2 3 1 4 9 8
3rd smallest element: 3
5th largest element: 4
```

6. Write a java program to do

i. Matrix addition

ii. Matrix multiplication

iii. Transpose of the given matrix

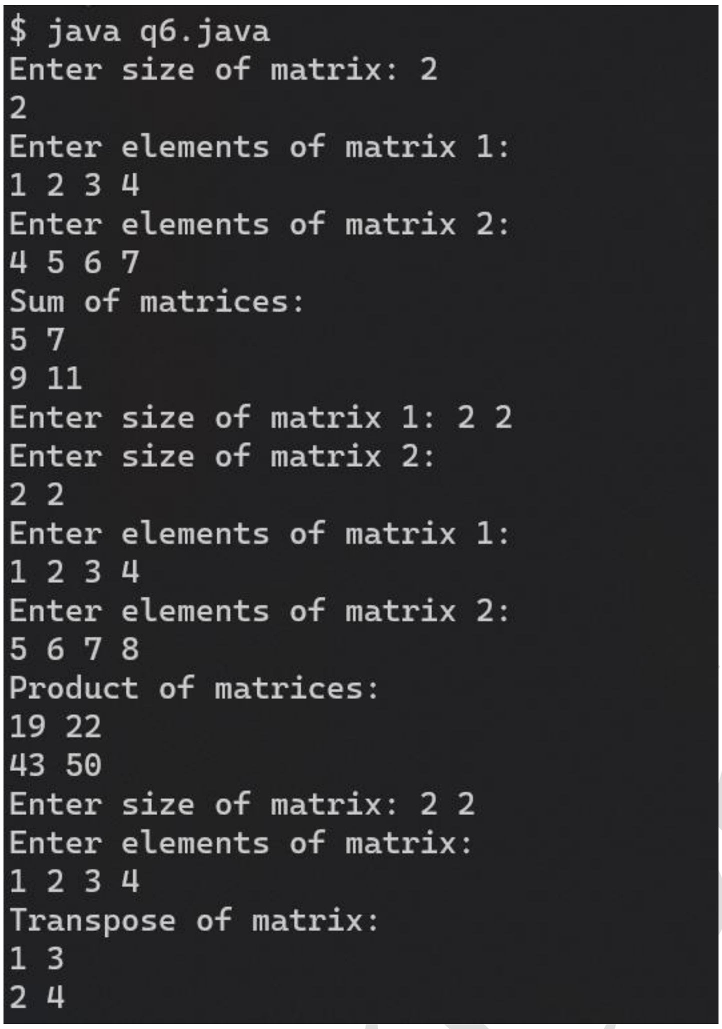
Source Code

```
import java.util.Scanner;


public class q6 {


    static void matrix_add() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter size of matrix: ");

        int n = sc.nextInt();

        int m = sc.nextInt();

        int mat1[][] = new int[n][m];

        int mat2[][] = new int[n][m];

        int mat3[][] = new int[n][m];

        System.out.println("Enter elements of matrix 1: ");

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

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

                mat1[i][j] = sc.nextInt();

            }

        }

        System.out.println("Enter elements of matrix 2: ");

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

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

                mat2[i][j] = sc.nextInt();

            }

        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                mat3[i][j] = mat1[i][j] + mat2[i][j];
            }
        }
        System.out.println("Sum of matrices: ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                System.out.print(mat3[i][j] + " ");
            }
            System.out.println();
        }
    }

    static void matrix_multiply() {
        Scanner sc = new Scanner(System.in);
        int n, m;
        int mat1[][];
        int mat2[][];
        int mat3[][];
        System.out.print("Enter size of matrix 1: ");
        n = sc.nextInt();
        m = sc.nextInt();
        System.out.println("Enter size of matrix 2: ");
        int p = sc.nextInt();
        int q = sc.nextInt();
        if (m != p) {
```

```
    System.out.println("Matrix multiplication not possible");
    sc.close();
    return;
}
mat1 = new int[n][m];
mat2 = new int[p][q];
mat3 = new int[n][q];
System.out.println("Enter elements of matrix 1: ");
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        mat1[i][j] = sc.nextInt();
    }
}
System.out.println("Enter elements of matrix 2: ");
for (int i = 0; i < p; i++) {
    for (int j = 0; j < q; j++) {
        mat2[i][j] = sc.nextInt();
    }
}
for (int i = 0; i < n; i++) {
    for (int j = 0; j < q; j++) {
        mat3[i][j] = 0;
        for (int k = 0; k < m; k++) {
            mat3[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
```

```
        }
        System.out.println("Product of matrices: ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < q; j++) {
                System.out.print(mat3[i][j] + " ");
            }
            System.out.println();
        }
    }

    static void transpose() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter size of matrix: ");
        int n = sc.nextInt();
        int m = sc.nextInt();
        int mat1[][] = new int[n][m];
        int mat2[][] = new int[m][n];
        System.out.println("Enter elements of matrix: ");
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                mat1[i][j] = sc.nextInt();
            }
        }
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                mat2[j][i] = mat1[i][j];
            }
        }
        System.out.println("Transpose of matrix: ");
        for (int i = 0; i < m; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(mat2[i][j] + " ");
            }
            System.out.println();
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        matrix_add();
        matrix_multiply();
        transpose();
        sc.close();
    }
}
```

Output:



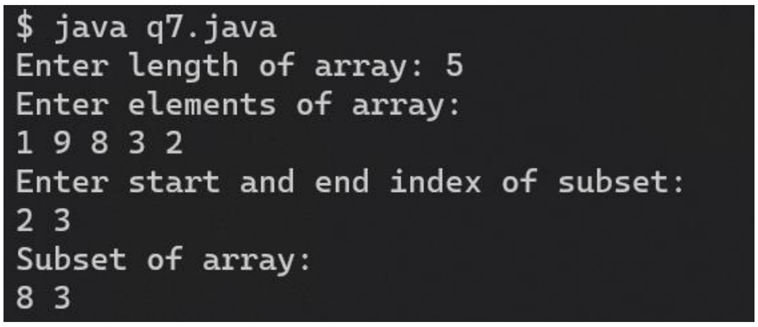
```
$ java q6.java
Enter size of matrix: 2
2
Enter elements of matrix 1:
1 2 3 4
Enter elements of matrix 2:
4 5 6 7
Sum of matrices:
5 7
9 11
Enter size of matrix 1: 2 2
Enter size of matrix 2:
2 2
Enter elements of matrix 1:
1 2 3 4
Enter elements of matrix 2:
5 6 7 8
Product of matrices:
19 22
43 50
Enter size of matrix: 2 2
Enter elements of matrix:
1 2 3 4
Transpose of matrix:
1 3
2 4
```

7. Write a java program to copy a subset of array elements to another array.

<u>Source Code</u>

```
import java.util.Scanner;


public class q7 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int n;

        System.out.print("Enter length of array: ");

        n = sc.nextInt();

        System.out.println("Enter elements of array: ");
        int arr[] = new int[n];
        for (int i = 0; i < n; i++) {
            arr[i] = sc.nextInt();
        }

        System.out.println("Enter start and end index of subset: ");
        int start = sc.nextInt();
        int end = sc.nextInt();

        int subset[] = new int[end - start + 1];
        System.arraycopy(arr, start, subset, 0, end - start + 1);

        System.out.println("Subset of array: ");
        for (int i = 0; i < end - start + 1; i++) {
            System.out.print(subset[i] + " ");
        }

        System.out.println();

        sc.close();

    }
}
```

Output:



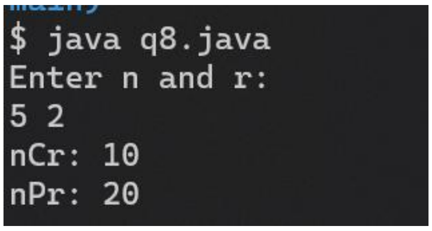
```
$ java q7.java
Enter length of array: 5
Enter elements of array:
1 9 8 3 2
Enter start and end index of subset:
2 3
Subset of array:
8 3
```

8. Write a Java program to calculate Permutation and Combination of 2 numbers.

Source Code

```
import java.util.Scanner;


public class q8 {
    static int factorial(int n) {
        if (n == 0)
            return 1;
        else
            return n * factorial(n - 1);
    }


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


        System.out.println("Enter n and r: ");
        int n = sc.nextInt();
        int r = sc.nextInt();
        int nCr = factorial(n) / (factorial(r) * factorial(n - r));
        System.out.println("nCr: " + nCr);
        int nPr = factorial(n) / factorial(n - r);
        System.out.println("nPr: " + nPr);


        sc.close();
    }
}
```

Output:



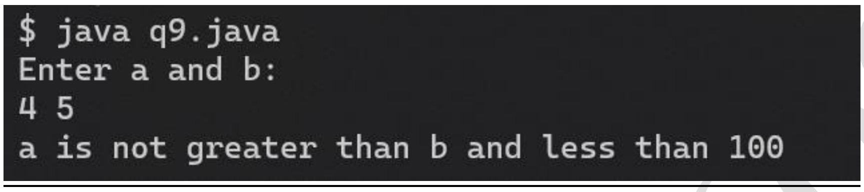
```
$ java q8.java
Enter n and r:
5 2
nCr: 10
nPr: 20
```

9. Write a program to enter the values of two variables 'a' and 'b' from keyboard and then check if both the conditions 'a < 100' and 'a > b' are true.

Source Code

```
import java.util.Scanner;


public class q9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);


        System.out.println("Enter a and b: ");
        int a = sc.nextInt();
        int b = sc.nextInt();


        if (a > b && a < 100) {
            System.out.println("a is greater than b and less than 100");
        } else {
            System.out.println("a is not greater than b and less than 100");
        }

        sc.close();
    }
}
```

Output:



```
$ java q9.java
Enter a and b:
4 5
a is not greater than b and less than 100
```

10. Write a java program with a class LibraryBooks with the following methods

InsertBook

BorrowBook

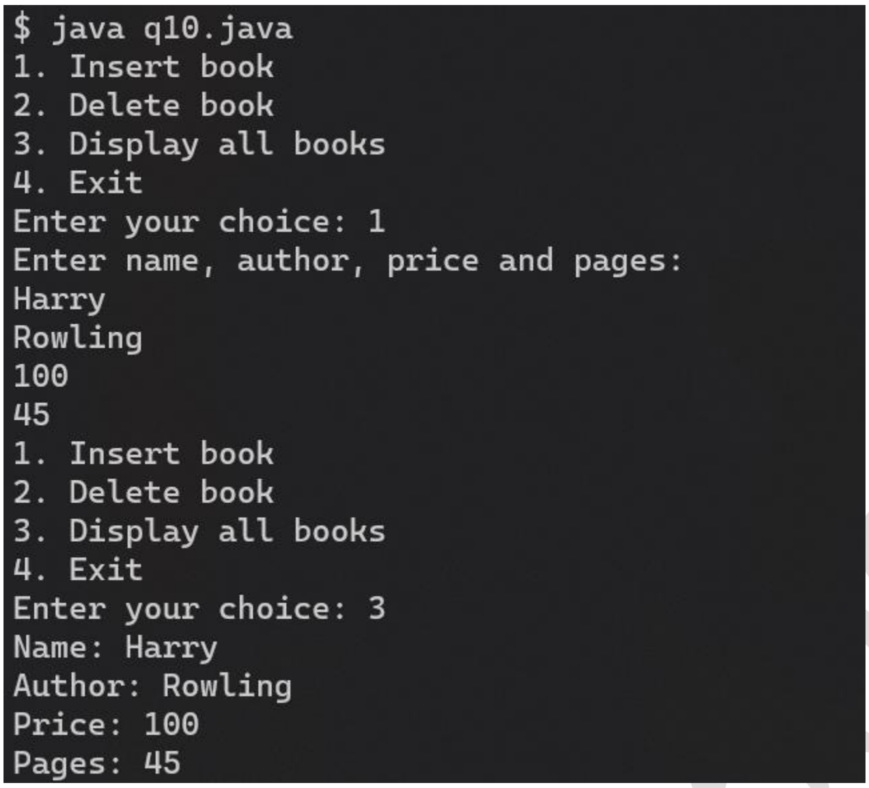
Display

Source Code

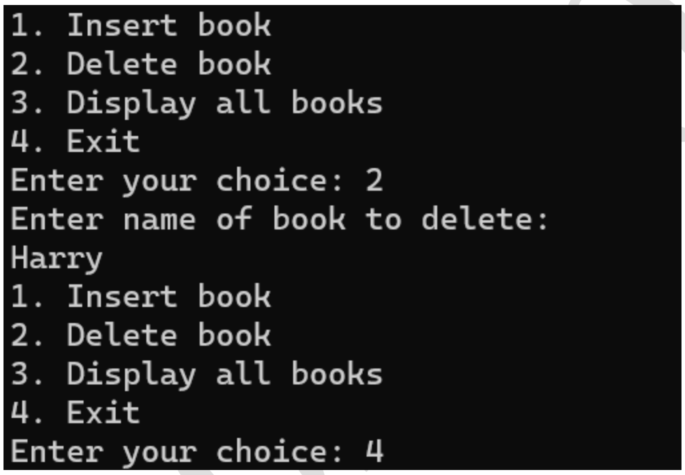
```
import java.util.Scanner;
public class q10 {
    static class Book {
        String name;
        String author;
        int price;
        int pages;
        Book(String name, String author, int price, int pages) {
            this.name = name;
            this.author = author;
            this.price = price;

            this.pages = pages;

        }

        void display() {

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

            System.out.println("Author: " + author);

            System.out.println("Price: " + price);

            System.out.println("Pages: " + pages);

        }

    }

    static Book[] books = new Book[10];

    static int count = 0;

    static void insertBook() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter name, author, price and pages: ");

        String name = sc.nextLine();

        String author = sc.nextLine();

        int price = sc.nextInt();

        int pages = sc.nextInt();

        books[count] = new Book(name, author, price, pages);

        count++;

    }


    static void deleteBook() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter name of book to delete: ");
    String name = sc.nextLine();
    for (int i = 0; i < count; i++) {
        if (books[i].name.equals(name)) {
            for (int j = i; j < count - 1; j++) {
                books[j] = books[j + 1];
            }
            count--;
            break;
        }
    }
}

static void displayBooks() {
    for (int i = 0; i < count; i++) {
        books[i].display();
    }
}
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    while (true) {
        System.out.println("1. Insert book");
        System.out.println("2. Delete book");
        System.out.println("3. Display all books");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
```

```
            int choice = sc.nextInt();


            switch (choice) {
                case 1:
                    insertBook();
                    break;
                case 2:
                    deleteBook();
                    break;
                case 3:
                    displayBooks();
                    break;
                case 4:
                    sc.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid choice");
            }
        }
    }
}
```

Output:



```
$ java q10.java
1. Insert book
2. Delete book
3. Display all books
4. Exit
Enter your choice: 1
Enter name, author, price and pages:
Harry
Rowling
100
45
1. Insert book
2. Delete book
3. Display all books
4. Exit
Enter your choice: 3
Name: Harry
Author: Rowling
Price: 100
Pages: 45
```



```
1. Insert book
2. Delete book
3. Display all books
4. Exit
Enter your choice: 2
Enter name of book to delete:
Harry
1. Insert book
2. Delete book
3. Display all books
4. Exit
Enter your choice: 4
```

Result:

Thus, basic java programs has been implemented successfully.

Exp. No: 3	CONSTRUCTOR, METHOD OVERLOADING, ARRAY OF OBJECTS
Date: 7.9.22	

Aim:

To implement constructor, method overloading and array of objects in java.

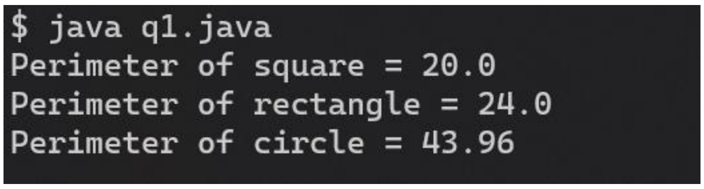
Program:

1. Write a program to compute perimeter of class Circle, Rectangle, Square using parameterized constructor.

Source Code:

```
public class q1 {


    static class Circle{
        double peri;

        Circle(double x){
            peri = 3.14 * 2 * x;
        }
    }

    static class Rectangle{
        double peri;

        Rectangle(double x, double y){
            peri = 2*(x+y);
        }
    }
    static class Square{
        double peri;

        Square(double x){
            peri = 4* x;
        }
    }

    public static void main(String args[]){
        Circle a = new Circle(7);
        Rectangle b = new Rectangle(7,5);
        Square c = new Square(5);

        System.out.println("Perimeter of square = "+c.peri);
        System.out.println("Perimeter of rectangle = "+b.peri);
        System.out.println("Perimeter of circle = "+a.peri);

    }
}
```

Output:



```
$ java q1.java
Perimeter of square = 20.0
Perimeter of rectangle = 24.0
Perimeter of circle = 43.96
```

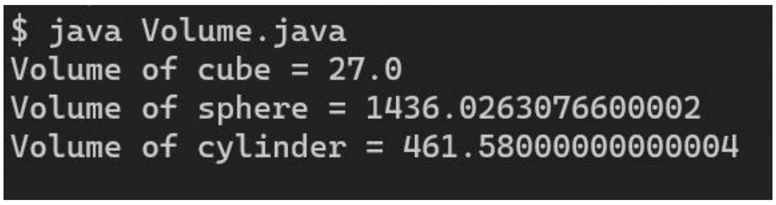
2. Write a program to design a class Volume and then find out the volume of a Cube, Cylinder and Sphere using method overloading.

Source Code:

```
public class Volume {

    double volume;

    Volume(int a) {
        volume = a * a * a;
    }

    Volume(double a) {
        volume = 1.333333 * 3.14 * a * a * a;
    }

    Volume(double r, int h) {
        volume = 3.14 * r * r * h;
    }

    public static void main(String args[]) {
        Volume a = new Volume(3);
        Volume b = new Volume(7.0);
        Volume c = new Volume(7.0, 3);

        System.out.println("Volume of cube = " + a.volume);
        System.out.println("Volume of sphere = " + b.volume);
        System.out.println("Volume of cylinder = " + c.volume);

    }
```

}

Output:



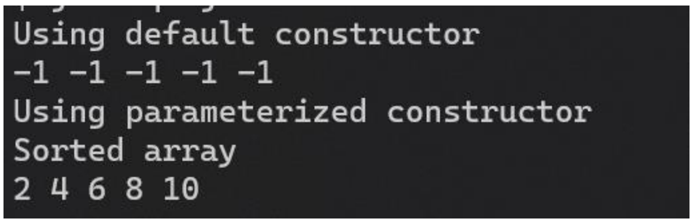
```
$ java Volume.java
Volume of cube = 27.0
Volume of sphere = 1436.0263076600002
Volume of cylinder = 461.58000000000004
```

3. Write a java class which consists of 5 integer data. Overload constructor (Default & parameterized) to initialize those integer data members. Write a method which sorts those integer data members using insertion sort.

Source Code

```
public class q3 {


    int[] array = new int[5];


    q3() {
        for (int i = 0; i < array.length; i++) {
            array[i] = -1;
        }
    }


    q3(int[] newArray) {
        System.arraycopy(newArray, 0, array, 0, array.length);
    }
    void sort() {
        for (int i = 1; i < array.length; i++) {
            int key = array[i];
            int j = i - 1;
            while (j >= 0 && array[j] > key) {
                array[j + 1] = array[j];
                j--;
            }
            array[j + 1] = key;
        }
    }
    void print() {
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }
    public static void main(String args[]) {
        System.out.println("Using default constructor");
        q3 x = new q3();
        x.print();
        int[] newArray;
        newArray = new int[] { 6, 4, 2, 8, 10 };
        System.out.println("Using parameterized constructor");
        q3 y = new q3(newArray);
        y.sort();
        System.out.println("Sorted array");
        y.print();

    }
}
```

Output:



```
Using default constructor
-1 -1 -1 -1 -1
Using parameterized constructor
Sorted array
2 4 6 8 10
```

4. Suppose you have a money box with an initial amount of Rs.500 and you have to add some more amount to it. Create a class “AddAmount” with a data member named “amount” with an initial value of Rs.500. Now create two constructors of this class as follows:

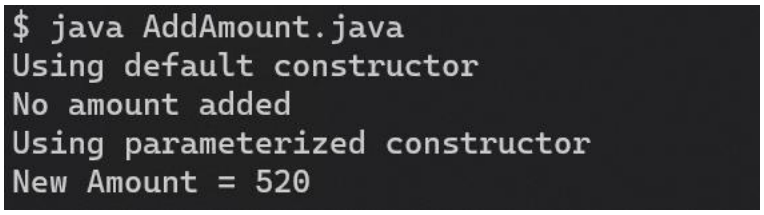
a. Without any parameter- no amount will be added to the money box

b. Having a parameter which is the amount that will be added to money box
Create an object of the “AddAmount” class and display the final amount in money box

Source Code

```
public class AddAmount {
    int amount = 500;


    AddAmount() {
        System.out.println("No amount added");
    }


    AddAmount(int x) {
        amount += x;
        System.out.println("New Amount = " + amount);
    }


    public static void main(String args[]) {
        System.out.println("Using default constructor");
        AddAmount x = new AddAmount();
        System.out.println("Using parameterized constructor");
        AddAmount y = new AddAmount(20);

    }

}
```

Output:



```
$ java AddAmount.java
Using default constructor
No amount added
Using parameterized constructor
New Amount = 520
```

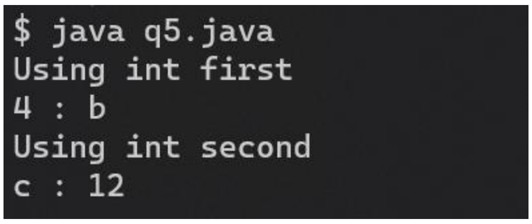
5. Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters.

For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).

Source Code

```
public class q5 {
    static void print(int n, char c) {
        System.out.println(n + " : " + c);
    }
    static void print(char c, int n) {
        System.out.println(c + " : " + n);
    }
    public static void main(String args[]) {
        System.out.println("Using int first");
        print(4, 'b');
        System.out.println("Using int second");
        print('c', 12);
    }
}
```

Output:



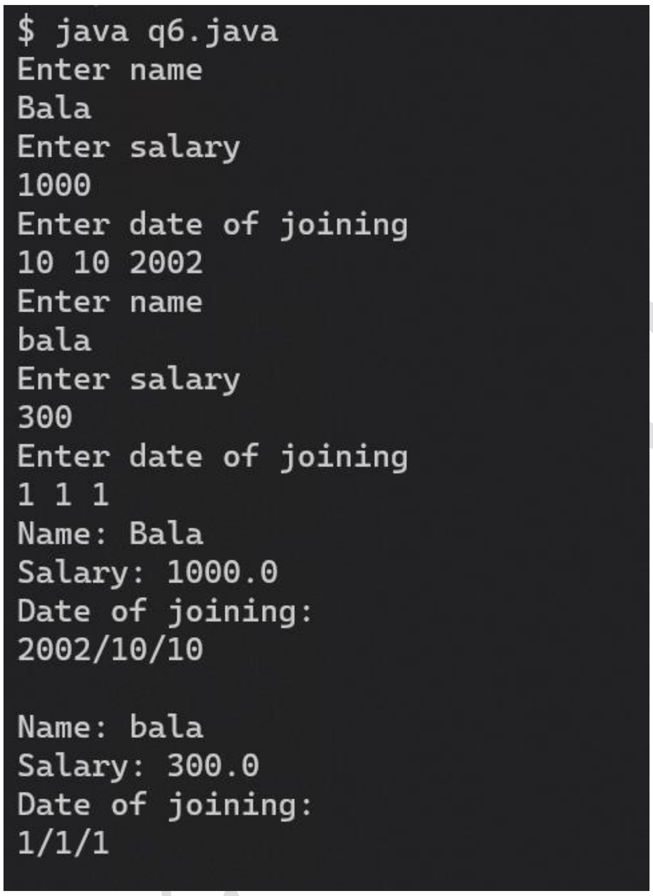
```
$ java q5.java
Using int first
4 : b
Using int second
c : 12
```

6. Write a program to print the name, salary and date of joining of 10 employees in a company (Note: Use array of objects.)

Source Code

```
import java.util.Date;

import java.util.Scanner;

public class q6 {

    static class DateClass {

        int day, month, year;

        DateClass(int d, int m, int y) {

            day = d;

            month = m;

            year = y;

        }

        void print() {

            System.out.println(day + "/" + month + "/" + year);

        }

    }

    static class Employee {

        String name;

        double salary;
        DateClass date;
    }
    static void print(Employee[] employees) {
        for (int i = 0; i < 2; i++) {
            System.out.println("Name: " + employees[i].name);
            System.out.println("Salary: " + employees[i].salary);
            System.out.println("Date of joining: ");
            employees[i].date.print();
            System.out.println();
        }
    }

    public static void main(String args[]) {
        Employee[] employees = new Employee[10];
        Scanner sc = new Scanner(System.in);
        int year, month, day;
        for (int i = 0; i < 2; i++) {
            employees[i] = new Employee();
            System.out.println("Enter name");
            employees[i].name = sc.nextLine();
            System.out.println("Enter salary");
            employees[i].salary = sc.nextDouble();
            System.out.println("Enter date of joining");
            year = sc.nextInt();
            month = sc.nextInt();
            day = sc.nextInt();
            employees[i].date = new DateClass(day, month, year);
            sc.nextLine();
        }
        print(employees);
    }
}
```

Output:



```
$ java q6.java
Enter name
Bala
Enter salary
1000
Enter date of joining
10 10 2002
Enter name
bala
Enter salary
300
Enter date of joining
1 1 1
Name: Bala
Salary: 1000.0
Date of joining:
2002/10/10

Name: bala
Salary: 300.0
Date of joining:
1/1/1
```

Result:

Thus, constructors, method overloading and array of objects has been implemented successfully.

Exp. No: 4	Static, This, Arrays, String Class
Date: 10.9.22	

Aim:

To implement Static, This, Arrays, String Class.

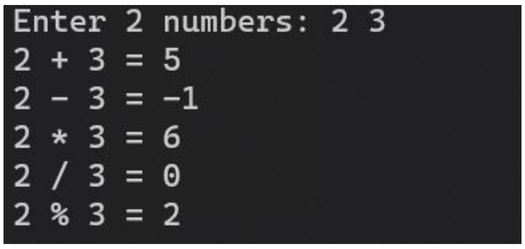
Program:

1. Write a java program to perform arithmetic (addition, subtraction, multiplication, division, modulo) operations using static members.

Source Code:

```
import java.util.Scanner;


public class q1 {

    static int a, b;


    static void add() {

        System.out.println(a + " + " + b + " = " + (a + b));

    }


    static void sub() {

        System.out.println(a + " - " + b + " = " + (a - b));

    }


    static void mul() {

        System.out.println(a + " * " + b + " = " + (a * b));

    }
    static void div() {
        System.out.println(a + " / " + b + " = " + (a / b));
    }

    static void mod() {
        System.out.println(a + " % " + b + " = " + (a % b));
    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.print("Enter 2 numbers: ");

        a = sc.nextInt();
        b = sc.nextInt();
        add();
        sub();
        mul();
        div();
        mod();
        sc.close();

    }
}
```

Output:



```
Enter 2 numbers: 2 3
2 + 3 = 5
2 - 3 = -1
2 * 3 = 6
2 / 3 = 0
2 % 3 = 2
```

2.. Write a Java program with a class BankAccount with the following methods

Credit

Debit

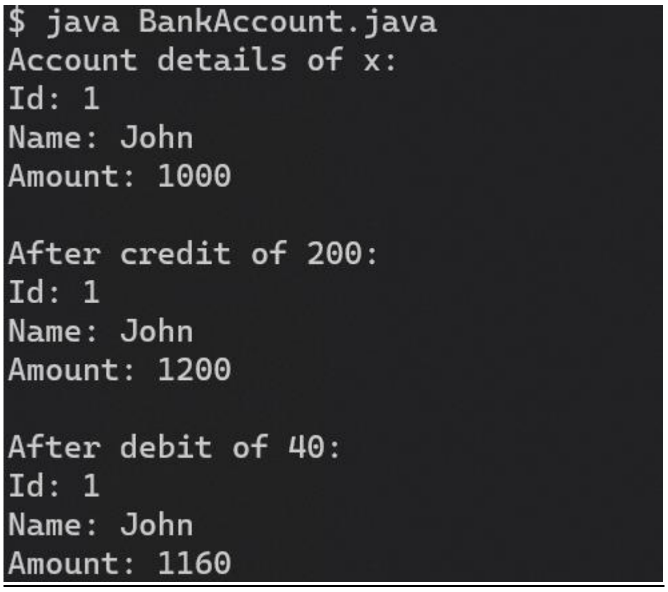
Display

And also illustrate the use of this keyword for the above scenario.

Source Code:

```
public class BankAccount {
    int id;
    String name;
    int amount;

    void credit(int amount) {
        this.amount += amount;
    }

    void debit(int amount) {
        if (this.amount < amount) {
            System.out.println("Amount withdrawn exceeds the current balance!");
        } else {
            this.amount -= amount;
        }
    }
    void display() {
        System.out.println("Id: " + id);
        System.out.println("Name: " + name);
        System.out.println("Amount: " + amount);
        System.out.println();
    }

    public static void main(String[] args) {
        BankAccount x = new BankAccount();
        System.out.println("Account details of x:");
        x.id = 1;
        x.name = "John";
        x.amount = 1000;
        x.display();

        System.out.println("After credit of 200:");
        x.credit(200);
        x.display();

        System.out.println("After debit of 40:");
        x.debit(40);
        x.display();
    }
}
```

Output:



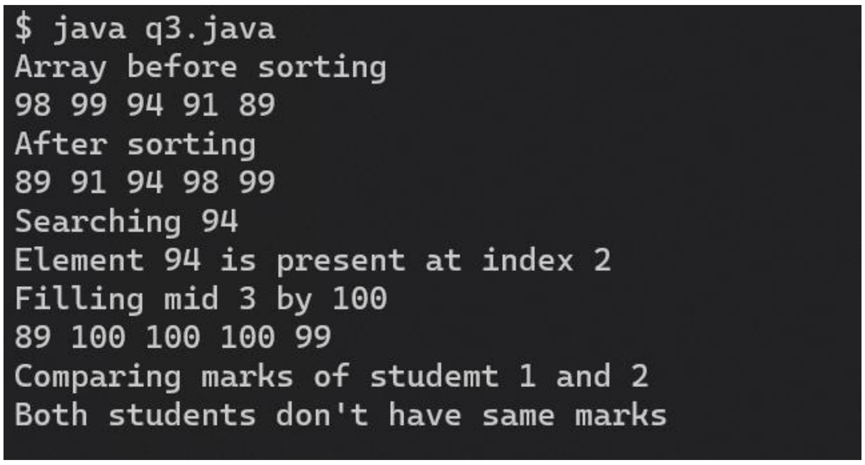
```
$ java BankAccount.java
Account details of x:
Id: 1
Name: John
Amount: 1000

After credit of 200:
Id: 1
Name: John
Amount: 1200

After debit of 40:
Id: 1
Name: John
Amount: 1160
```

3. Write a Java program to store the marks of the student in an array and using Arrays Class methods (sort,fill,Search,equals) perform the manipulations.

<u>Source Code</u>

```
import java.util.Arrays;


public class q3 {


    static void print(int[] array){
        for(int i=0;i<array.length;i++){
            System.out.print(array[i]+ " ");
        }
        System.out.println("");
    }




    public static void main (String[] args){
        int marks1 [] = new int[]{98, 99, 94, 91, 89};
        int marks2 [] = new int[]{98, 99, 94, 91, 84};


        System.out.println("Array before sorting");

        print(marks1);


        Arrays.sort(marks1);

        System.out.println("After sorting");

        print(marks1);


        System.out.println("Searching 94");

        System.out.println("Element 94 is present at index "+ Arrays.binarySearch(marks1,
94));


        Arrays.fill(marks1, 1, 4, 100);

        System.out.println("Filling mid 3 by 100");

        print(marks1);


        System.out.println("Comparing marks of studemt 1 and 2");

        if(Arrays.equals(marks1, marks2)){

           System.out.println("Both students have same marks");

        }

        else{

           System.out.println("Both students don't have same marks");

        }

    }

}
```

Output:



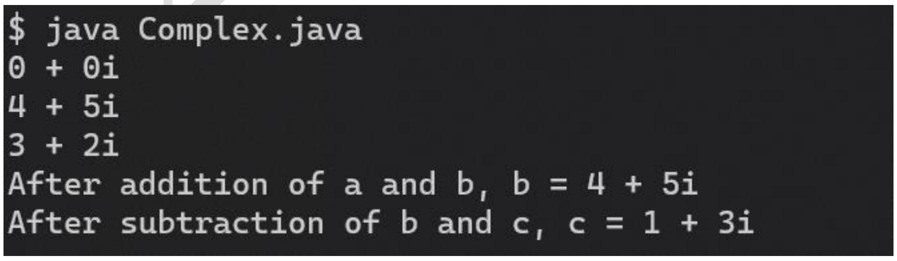
```
$ java q3.java
Array before sorting
98 99 94 91 89
After sorting
89 91 94 98 99
Searching 94
Element 94 is present at index 2
Filling mid 3 by 100
89 100 100 100 99
Comparing marks of studemt 1 and 2
Both students don't have same marks
```

4. Write a program to display complex number using constructor overloading, and also perform simple arithmetic operation with complex numbers using this keyword along with the variables.

<u>Source Code</u>

```
public class Complex {

    int r, c;

    Complex(){

        r = 0; c= 0;

        System.out.println(r+ " + "+ c+ "i");

    }

    Complex(int r, int c){

        this.r = r;

        this.c = c;


        System.out.println(r+ " + "+ c+ "i");

    }


    void add(Complex a, Complex b){

        this.r = a.r + b.r;

        this.c = a.c + b.c;
    }

    void sub(Complex a, Complex b){
        this.r = a.r - b.r;
        this.c = a.c - b.c;
    }

    public static void main(String[] args){
        Complex a= new Complex();
        Complex b = new Complex(4, 5);
        Complex c = new Complex(3, 2);

        b.add(a, b);
        System.out.println("After addition of a and b, b = " + b.r+ " + "+ b.c+ "i");
        c.sub(b, c);
        System.out.println("After subtraction of b and c, c = " + c.r+ " + "+ c.c+ "i");
    }

}
```

Output:



```
$ java Complex.java
0 + 0i
4 + 5i
3 + 2i
After addition of a and b, b = 4 + 5i
After subtraction of b and c, c = 1 + 3i
```

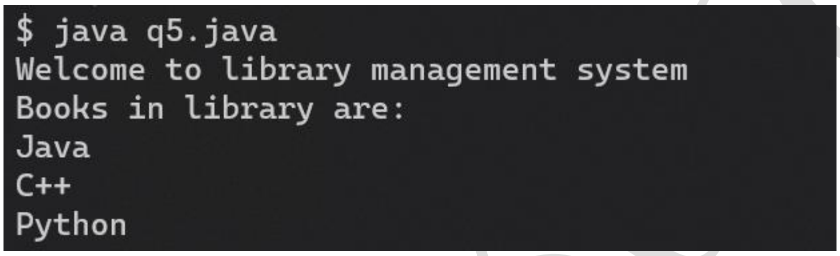
5. Write a program to implement library management with static methods, static variables

and static blocks.

Source Code

```
public class q5 {

    static int bookCount = 0;
    static String[] books = new String[10];

    static {
        System.out.println("Welcome to library management system");
    }

    static void addBook(String book) {
        if (bookCount < 10) {
            books[bookCount] = book;
            bookCount++;
        } else {
            System.out.println("No more books can be added");
        }
    }

    static void displayBooks() {
        System.out.println("Books in library are: ");
        for (int i = 0; i < bookCount; i++) {
            System.out.println(books[i]);
        }
    }
    public static void main(String[] args) {
        addBook("Java");
        addBook("C++");
        addBook("Python");
        displayBooks();
    }
}
```

Output:



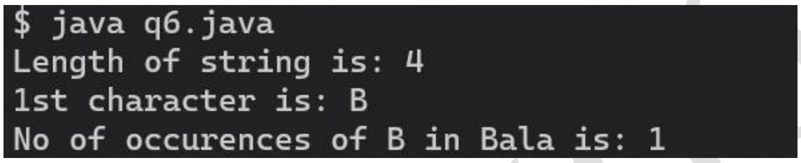
```
$ java q5.java
Welcome to library management system
Books in library are:
Java
C++
Python
```

6. Write a Java program to create a string object. Initialize this object with your name. Find the length of your name using appropriate method. Find the 1st character of your name and find the number of time the character appears in your name.

Source Code

```
public class q6 {
    public static void main(String[] args) {
        String s = "Bala";
        System.out.println("Length of string is: " + s.length());
        char b = s.charAt(0);
        System.out.println("1st character is: " + b);

        // find no of occurences of b in s
        int count = 0;
        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == b) {
                count++;
            }
        }
        System.out.println("No of occurences of " + b + " in " + s + " is: " + count);

    }
}
```

Output:



```
$ java q6.java
Length of string is: 4
1st character is: B
No of occurences of B in Bala is: 1
```

7. Write program in Java for String handling (StringBuffer, StringBuilder) which performs the following:

i. Checks the capacity.

ii. Reverse the contents of a string.

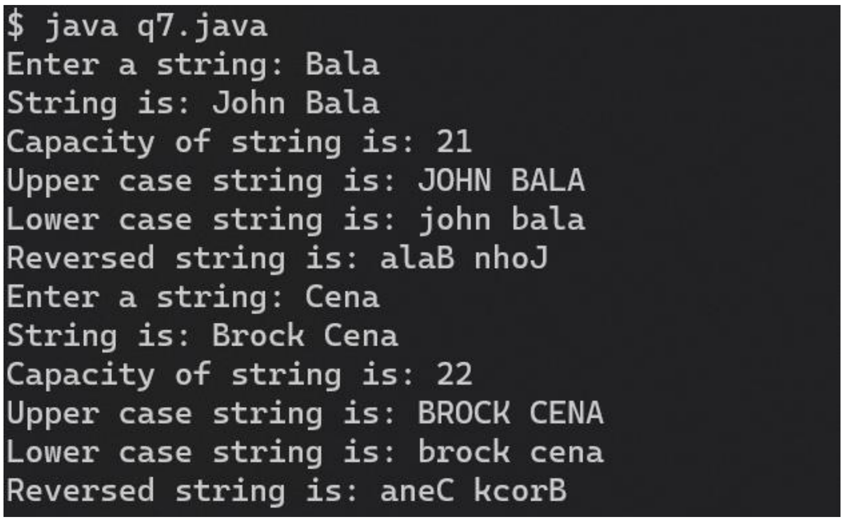
iii. Convert the string in upper case/lower case.

iv. Read a string from user and append it.

Source Code

```
import java.util.Scanner;


public class q7 {
    public static void main(String[] args) {
        StringBuffer s = new StringBuffer("John ");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        s.append(sc.nextLine());
        System.out.println("String is: " + s);
        System.out.println("Capacity of string is: " + s.capacity());
        System.out.println("Upper case string is: " + s.toString().toUpperCase());
        System.out.println("Lower case string is: " + s.toString().toLowerCase());
        // reverse the string
        System.out.println("Reversed string is: " + s.reverse());

        StringBuilder s1 = new StringBuilder("Brock ");
        System.out.print("Enter a string: ");

        s1.append(sc.nextLine());

        System.out.println("String is: " + s1);
        System.out.println("Capacity of string is: " + s1.capacity());
        System.out.println("Upper case string is: " + s1.toString().toUpperCase());
        System.out.println("Lower case string is: " + s1.toString().toLowerCase());

        // reverse the string
        System.out.println("Reversed string is: " + s1.reverse());

        sc.close();
    }
}
```

Output:



```
$ java q7.java
Enter a string: Bala
String is: John Bala
Capacity of string is: 21
Upper case string is: JOHN BALA
Lower case string is: john bala
Reversed string is: alaB nhoJ
Enter a string: Cena
String is: Brock Cena
Capacity of string is: 22
Upper case string is: BROCK CENA
Lower case string is: brock cena
Reversed string is: aneC kcorB
```

Result:

Thus, static, this, arrays and string class has been implemented successfully.

Exp. No: 5	Inheritance, Super, Final and ArrayList
Date: 14.9.22	

Aim:

To implement inheritance, super, final and array list.

Program:

1. Create a class named 'Member' having the following members: Data members

1 - Name

2 - Age

3 - Phone number

4 - Address

5 – Salary

It also has a method named 'printSalary' which prints the salary of the members.

Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same. (Note: Use all access specifier)

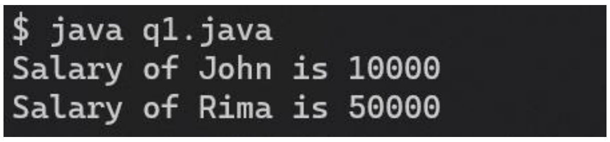
Source Code:

```
public class q1 {


    static class member {

        String name;

        int age;

        String phone;

        String address;

        int salary;


        void printSalary() {
```

```
        System.out.println("Salary of " + name + " is " + salary);
    }
}

static class employee extends member {
    String specialization;
}

static class manager extends member {
    String department;
}

public static void main(String[] args) {

    employee e = new employee();
    e.name = "John";
    e.age = 20;
    e.phone = "9876543210";
    e.address = "Chennai";
    e.salary = 10000;
    e.specialization = "Java";
    e.printSalary();

    manager m = new manager();
    m.name = "Rima";
    m.age = 32;
```

```
        m.phone = "9876543201";

        m.address = "Bangalore";

        m.salary = 50000;

        m.department = "Management";

        m.printSalary();


    }

}
```

Output:



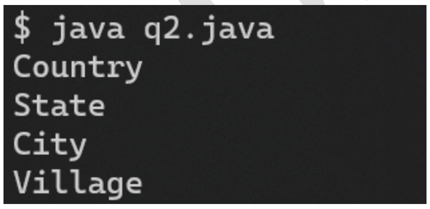
```
$ java q1.java
Salary of John is 10000
Salary of Rima is 50000
```

2. Create a class country, state, city and village. Arrange these classes in hierarchical manner.

Source Code:

```
public class q2 {

    static class country {

        country() {

            System.out.println("Country");

        }

    }


    static class state extends country {

        state() {

            System.out.println("State");

        }

    }
    static class city extends state {
        city() {
            System.out.println("City");
        }
    }

    static class village extends city {
        village() {
            System.out.println("Village");
        }
    }

    public static void main(String[] args) {
        village v = new village();
    }
}
```

Output:



```
$ java q2.java
Country
State
City
Village
```

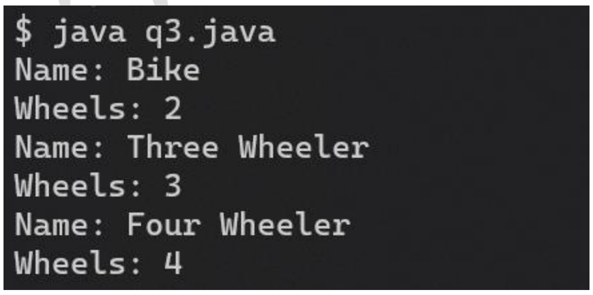
3. Declare a class of vehicle. Derived classes are two-wheeler, three-wheeler and four-wheeler. Display the properties of each type of vehicle using member function of class.

Source Code

```
public class q3 {
```

```
static class vehicle {
    String name;
    int wheels;

    void print() {
        System.out.println("Name: " + name);
        System.out.println("Wheels: " + wheels);
    }
}

static class two_wheeler extends vehicle {
    two_wheeler() {
        name = "Bike";
        wheels = 2;
    }
}

static class three_wheeler extends vehicle {
    three_wheeler() {
        name = "Three Wheeler";
        wheels = 3;
    }
}

static class four_wheeler extends vehicle {
    four_wheeler() {
            name = "Four Wheeler";
            wheels = 4;
        }
    }

    public static void main(String[] args) {

        two_wheeler t = new two_wheeler();
        t.print();

        three_wheeler th = new three_wheeler();
        th.print();

        four_wheeler f = new four_wheeler();
        f.print();

    }
}
```

Output:



```
$ java q3.java
Name: Bike
Wheels: 2
Name: Three Wheeler
Wheels: 3
Name: Four Wheeler
Wheels: 4
```

4. Write a Java program to implement the following scenario with these keywords/concepts listed below: i.) this ii.)super iii.)final iv.)Access Modifier v.) static

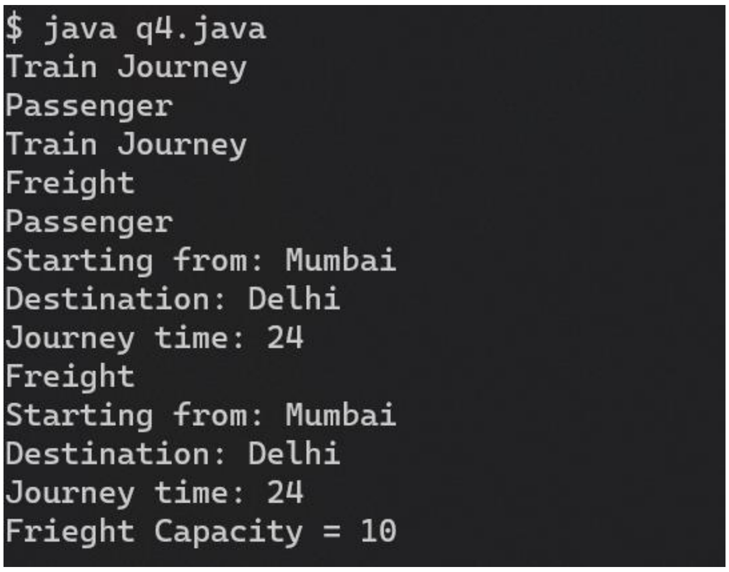
Source Code

```
public class q4 {

    static class trainJourney {
        String starting_from;
        String destination;
        int j_time;

        trainJourney() {
            System.out.println("Train Journey");
        }

        void set_starting_from(String s) {
            starting_from = s;
        }

        void set_destination(String d) {
            destination = d;
        }

        void set_j_time(int j_time) {
            this.j_time = j_time;
        }
    }

    // freight extends train
    static class freight extends trainJourney {
```

```
    freight() {
        super();
        System.out.println("Freight");
    }

    final int capacity = 10;
    int carriages;
    boolean hazard;

    int get_capacity() {
        return capacity;
    }

    void set_hazard(boolean h) {
        hazard = h;
    }
}

// passenger extends train
static class passenger extends trainJourney {
    passenger() {
        super();
        System.out.println("Passenger");
    }

    int fc;
    int carraiges;
    boolean catergory;

    int get_fc() {
        return fc;
    }

    // add carriages
    void add_carriages(int c) {
        carraiges += c;
    }

    boolean get_catergory() {
        return catergory;
    }
}

public static void main(String[] args) {

    // create passenger variable using super
    passenger p = new passenger();
    p.set_starting_from("Mumbai");
    p.set_destination("Delhi");
    p.set_j_time(24);

    // create freight variable using super
```

```
        freight f = new freight();
        f.set_starting_from("Mumbai");
        f.set_destination("Delhi");
        f.set_j_time(24);

        // print passenger
        System.out.println("Passenger");
        System.out.println("Starting from: " + p.starting_from);
        System.out.println("Destination: " + p.destination);
        System.out.println("Journey time: " + p.j_time);

        // print freight
        System.out.println("Freight");
        System.out.println("Starting from: " + f.starting_from);
        System.out.println("Destination: " + f.destination);
        System.out.println("Journey time: " + f.j_time);
        System.out.println("Frieght Capacity = " + f.capacity);

    }
}
```

Output:

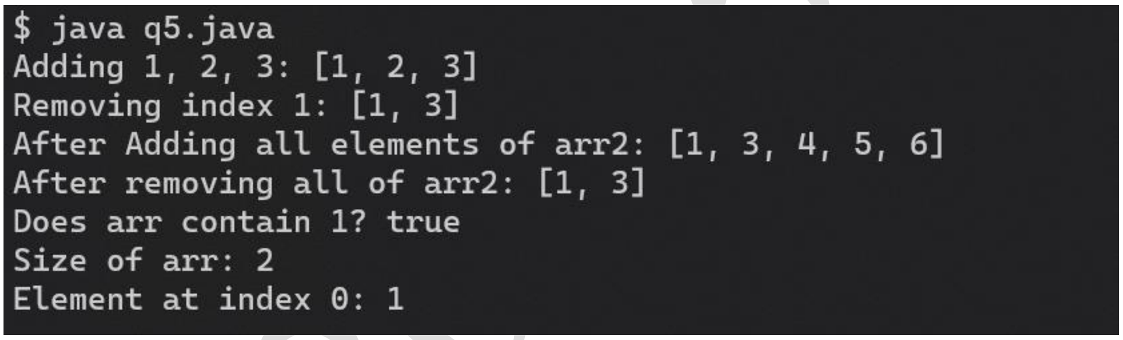

```
$ java q4.java
Train Journey
Passenger
Train Journey
Freight
Passenger
Starting from: Mumbai
Destination: Delhi
Journey time: 24
Freight
Starting from: Mumbai
Destination: Delhi
Journey time: 24
Frieght Capacity = 10
```

5. Write a java program to store details of student and using ArrayList class methods (add,remove,addall,remove,contains,size,get) do the necessary manipulations.

<u>Source Code</u>

```
import java.util.ArrayList;


public class q5 {
    public static void main(String[] args) {
        ArrayList<Integer> arr = new ArrayList<Integer>();
        arr.add(1);
        arr.add(2);
        arr.add(3);
        System.out.println("Adding 1, 2, 3: " + arr);
        arr.remove(1);
        System.out.println("Removing index 1: " + arr);
        ArrayList<Integer> arr2 = new ArrayList<Integer>();
        arr2.add(4);
        arr2.add(5);
        arr2.add(6);

        arr.addAll(arr2);

        System.out.println("After Adding all elements of arr2: " + arr);

        arr.removeAll(arr2);

        System.out.println("After removing all of arr2: " + arr);

        System.out.println("Does arr contain 1? " + arr.contains(1));

        System.out.println("Size of arr: " + arr.size());

        System.out.println("Element at index 0: " + arr.get(0));

    }

}
```

Output:



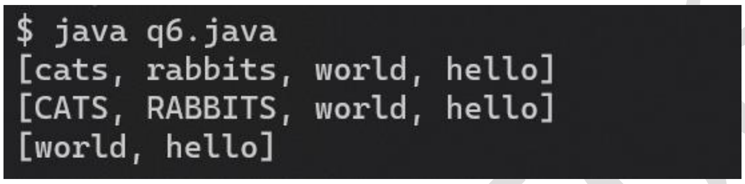
```
$ java q5.java
Adding 1, 2, 3: [1, 2, 3]
Removing index 1: [1, 3]
After Adding all elements of arr2: [1, 3, 4, 5, 6]
After removing all of arr2: [1, 3]
Does arr contain 1? true
Size of arr: 2
Element at index 0: 1
```

6.. Write a method reverse that reverses the order of the elements in an ArrayList of strings. Write a method capitalizePlurals that accepts an ArrayList of strings and replaces every word ending with an "s" with its uppercased version. Write a method removePlurals that accepts an ArrayList of strings and removes every word in the list ending with an "s", case-insensitively.

Source Code

```
import java.util.*;


public class q6 {

    public static ArrayList<String> capitalizePlural(ArrayList<String> arr) {

        ArrayList<String> newArr = new ArrayList<String>();
        for (String i : arr) {
            if (i.charAt(i.length() - 1) == 's') {
                i = i.toUpperCase();
            }
            newArr.add(i);
        }
        return newArr;
    }

    public static ArrayList<String> removePlural(ArrayList<String> arr) {
        ArrayList<String> newArr = new ArrayList<String>();
        for (String i : arr) {
            if (i.charAt(i.length() - 1) != 's') {
                newArr.add(i);
            }
        }
        return newArr;
    }

    public static void main(String[] args) {
        ArrayList<String> arr = new ArrayList<String>();
        arr.add("hello");
        arr.add("world");
        arr.add("rabbits");
        arr.add("cats");
        Collections.reverse(arr);
        System.out.println(arr);

        ArrayList<String> arr2 = arr;

        arr = capitalizePlural(arr);

        System.out.println(arr);

        arr2 = removePlural(arr2);

        System.out.println(arr2);


    }

}
```

Output:



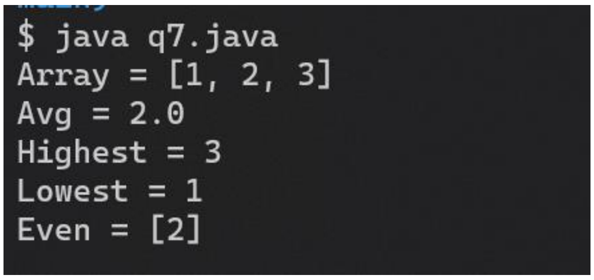
```
$ java q6.java
[cats, rabbits, world, hello]
[CATS, RABBITS, world, hello]
[world, hello]
```

7. Write a program that reads a numbers from ArrayList and displays all the numbers as a list, then: a. Prints the average of the numbers. b. Prints the highest and lowest number. c. Filters out all of the even numbers (ones divisible by 2).

Source Code

```
import java.util.*;


public class q7 {

    public static void main(String[] args) {

        ArrayList<Integer> arr = new ArrayList<Integer>();

        arr.add(1);

        arr.add(2);

        arr.add(3);

        System.out.println("Array = " + arr);

        ArrayList<Integer> newArr = new ArrayList<Integer>();
        double avg = 0;
        int highest = arr.get(0);
        int lowest = arr.get(0);
        for (int i : arr) {
            avg += i;
            if (i > highest) {
                highest = i;
            }
            if (i < lowest) {
                lowest = i;
            }

            if (i % 2 == 0) {
                newArr.add(i);
            }
        }
        avg /= arr.size();
        System.out.println("Avg = " + avg);
        System.out.println("Highest = " + highest);
        System.out.println("Lowest = " + lowest);
        System.out.println("Even = " + newArr);

    }
}
```

<u>Output:</u>



```
$ java q7.java
Array = [1, 2, 3]
Avg = 2.0
Highest = 3
Lowest = 1
Even = [2]
```

Result:

Thus, inheritance, super, final and arryalist has been implemented successfully.

Exp. No: 6	Inheritance, Abstract, Polymorphism, Regex
Date:21.9.22	

Aim:

To implement inheritance, abstract, polymorphism and regex in java

Program:

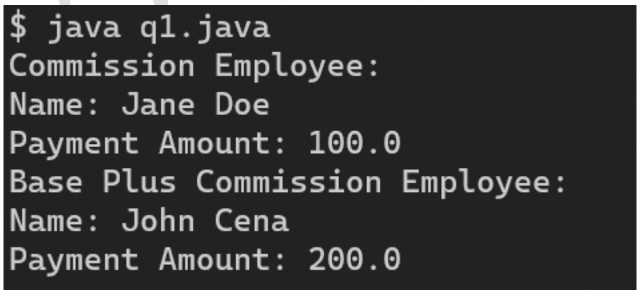
1. Write a Java program to implement the following specified in the picture using inheritance, interface concepts and calculate the salary of an employee. Use super keyword , access modifiers

Source Code:

```
public class q1 {
    interface payable {
        double getPaymentAmount();
    }
    static class Employee implements payable {
        String fname;
        String lname;
        String ssn;

        public double getPaymentAmount() {
            return 0;
        }
    }
    static class Invoice implements payable {
        String partNumber;
        String partDescription;
        int quantity;
    double pricePerItem;


    public double getPaymentAmount() {
        return quantity * pricePerItem;
    }
}
static class CommissionEmployee extends Employee {
    double grossSales;
    double commissionRate;


    public double getPaymentAmount() {
        return grossSales * commissionRate;
    }


}
static class HourlyEmployee extends Employee {
    double wage;
    double hours;


    public double getPaymentAmount() {
        return wage * hours;
    }
}
static class WeeklyEmployee extends Employee {
    double salary;
    public double getPaymentAmount() {
        return salary;
    }
}
static class BasePlusCommissionEmployee extends CommissionEmployee {
    double baseSalary;

    public double getPaymentAmount() {
        return baseSalary + super.getPaymentAmount();
    }
}

static void print(Object o) {
    System.out.println(o);
}

public static void main(String[] args) {

    CommissionEmployee ce = new CommissionEmployee();
    BasePlusCommissionEmployee bce = new BasePlusCommissionEmployee();

    ce.fname = "Jane";
    ce.lname = "Doe";
    ce.ssn = "987654321";
    ce.grossSales = 1000;
    ce.commissionRate = 0.1;
```

```
        bce.fname = "John";

        bce.lname = "Cena";

        bce.ssn = "123456789";

        bce.grossSales = 1000;

        bce.commissionRate = 0.1;

        bce.baseSalary = 100;


        print("Commission Employee:");

        print("Name: " + ce.fname + " " + ce.lname);

        print("Payment Amount: " + ce.getPaymentAmount());


        print("Base Plus Commission Employee:");

        print("Name: " + bce.fname + " " + bce.lname);

        print("Payment Amount: " + bce.getPaymentAmount());


    }

}
```

Output:



```
$ java q1.java
Commission Employee:
Name: Jane Doe
Payment Amount: 100.0
Base Plus Commission Employee:
Name: John Cena
Payment Amount: 200.0
```

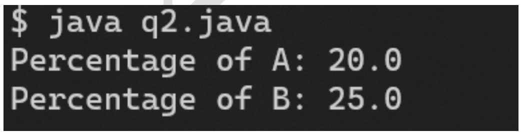
2. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The

constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

Source Code:

```
public class q2 {
    abstract static class Marks {
        abstract double getPercentage();
    }
    static class A extends Marks {
        double m1, m2, m3;

        A(double m1, double m2, double m3) {
            this.m1 = m1;
            this.m2 = m2;
            this.m3 = m3;
        }

        public double getPercentage() {
            return (m1 + m2 + m3) / 3;
        }
    }
    static class B extends Marks {
        double m1, m2, m3, m4;

        B(double m1, double m2, double m3, double m4) {
            this.m1 = m1;
            this.m2 = m2;
            this.m3 = m3;

            this.m4 = m4;

        }


        public double getPercentage() {

            return (m1 + m2 + m3 + m4) / 4;

        }

    }


    public static void main(String[] args) {


        A a = new A(10, 20, 30);

        B b = new B(10, 20, 30, 40);


        System.out.println("Percentage of A: " + a.getPercentage());

        System.out.println("Percentage of B: " + b.getPercentage());


    }

}
```

Output:



```
$ java q2.java
Percentage of A: 20.0
Percentage of B: 25.0
```

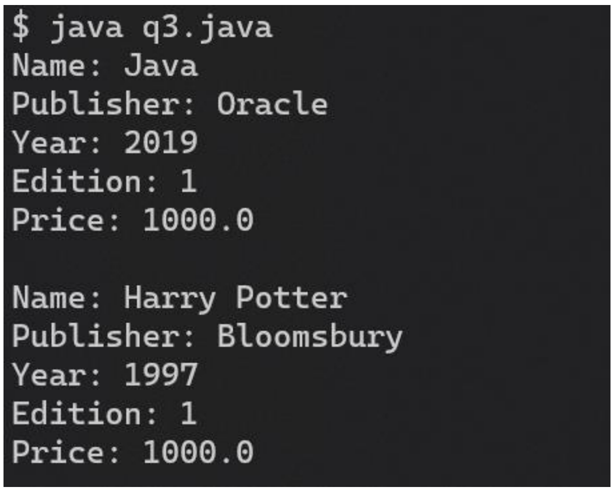
3. Write a program OnlinebookShop which sells and then add Technical Book and Non-Technical book as subclasses of the Book class which holds the details about book name, publisher, Published year, edition and price. Implement the polymorphic behavior through overriding methods. Explore the usage of Upcasting and Downcasting also.

Source Code

```
public class q3 {

    static class OnlineBook {
        String name;
        String publisher;
        int year;
        int edition;
        double price;
        public double getPrice() {
            return price;
        }

    }

    static class TechBook extends OnlineBook {
        String language;
        String framework;

        public double getPrice() {
            return price * 0.9;
        }
    }

    static class NonTechBook extends OnlineBook {
        String author;
        String genre;
        public double getPrice() {
            return price * 0.8;
        }
    }

    static void print(Object o) {
        System.out.println(o);
    }

    static void print(OnlineBook o) {
        print("Name: " + o.name);
        print("Publisher: " + o.publisher);
        print("Year: " + o.year);
        print("Edition: " + o.edition);
        print("Price: " + o.price);
    }

    public static void main(String[] args) {

        // upcasting
        OnlineBook ob = new TechBook();
        ob.name = "Java";
        ob.publisher = "Oracle";
        ob.year = 2019;
        ob.edition = 1;
```

```
        ob.price = 1000;

        OnlineBook ob2 = new NonTechBook();
        ob2.name = "Harry Potter";
        ob2.publisher = "Bloomsbury";
        ob2.year = 1997;
        ob2.edition = 1;
        ob2.price = 1000;

        // downcasting
        TechBook tb = (TechBook) ob;
        tb.language = "Java";
        tb.framework = "Spring";

        print(tb);
        print("");

        NonTechBook ntb = (NonTechBook) ob2;
        ntb.author = "J.K. Rowling";
        ntb.genre = "Fantasy";

        print(ntb);
    }
}
```

<u>Output:</u>


```
$ java q3.java
Name: Java
Publisher: Oracle
Year: 2019
Edition: 1
Price: 1000.0

Name: Harry Potter
Publisher: Bloomsbury
Year: 1997
Edition: 1
Price: 1000.0
```

4. Write a java program for registering the details of jobseeker. The requirements are mentioned below.

i. username should always end with _job and there should be atleast minimum of characters to the left of _job.

ii. Validate the emaild provided by the user.

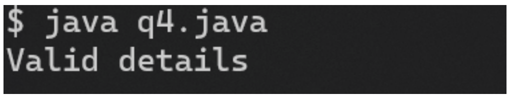
iii. Password should accept only characters and numbers.

Source Code:

```
import java.util.regex.*;


public class q4 {
    static void print(Object o) {
        System.out.println(o);
    }


    static class JobApplication {
        String name;
        String email;
        String password;


        JobApplication(String name, String email, String password) {
```

```
        this.name = name;

        this.email = email;

        this.password = password;

    }


    boolean validateEmail() {

        return Pattern.matches("^(.+)@(.+)$", email);

    }


    boolean validatePassword() {

        String regex = "^[a-zA-Z0-9]+$";

        return Pattern.matches(regex, password);

    }


    boolean validateName() {

        // ends with _job

        return Pattern.matches("^.+_job$", name);

    }


}


public static void main(String[] args) {

    JobApplication ja = new JobApplication("abc_job", "abc@mail.com", "abc123");

    if (ja.validateEmail() && ja.validatePassword() && ja.validateName()) {

        print("Valid details");

    } else {
            print("Invalid details");
        }


    }
}
```

Output:



```
$ java q4.java
Valid details
```

5. An company requires each employee to maintain a secret code. The secret code needs to pass certain validation for getting accepted. The validation rules are given as follows:

i. The secret code should be six characters long

ii. The first three characters should be cod

iii. There should be at least one digit in code (use isDigit)

iv. The first character should always be an upper case letter (Use isUpperCase)

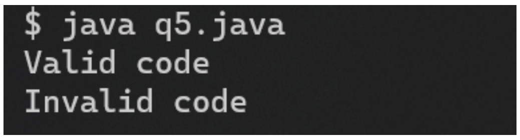
v. The code should contain only alphabets and digits

Return true if the above validation is passed.

Source Code:

```
public class q5 {


    static boolean validate_code(String s) {
        if (s.length() != 6) {
            return false;
        }
        if (!s.substring(0, 3).equals("Cod")) {
            return false;
        }
        if (!s.matches(".*\\d.*")) {
            return false;
        }
        if (!Character.isUpperCase(s.charAt(0))) {
            return false;
        }
        if (!s.matches("^[a-zA-Z0-9]+$")) {
            return false;
        }

        return true;
    }

    static void validate(String s) {
        if (validate_code(s)) {
            System.out.println("Valid code");
        } else {
            System.out.println("Invalid code");
        }
    }

    public static void main(String[] args) {
        String s = "Cod1e4";
        validate(s);

        s = "code";
        validate(s);


    }

}
```

Output:



```
$ java q5.java
Valid code
Invalid code
```

Result:

Thus, inheritance, abstract, polymorphism and regex has been implemented successfully.

Exp. No: 7	Exception Handling
Date: 28.9.22	

Aim:

To implement exception handling in java.

Program:

1. Develop a Java Console application to design a Vending Machine which follows following requirements

1. Accepts coins of ₹1, ₹5, ₹10, ₹25, ₹50
2. Allows user to select products (Chocolate(10), Snack(25), Nuts(50), Juice(20))
3. Allow user to take refund by cancelling the request
4. Return selected product and remaining change, if any
5. Allow reset operation for vending machine supplier

Use Interface, Inheritance, Constructor, abstract class, polymorphism, exception (Unchecked Exception, checked Exception, Custom Exception (NotPaidFullAmoutException, NoSufficientChangeException, SoldOutException)) concepts and also use static, final and this keyword wherever applicable to design a java application.

Source Code:

```
import java.util.*;

public class q1 {


   abstract static class ProductFn {


        int price;

        int quantity;

        String name;


        ProductFn(int price, int quantity, String name) {
            this.price = price;

            this.quantity = quantity;

            this.name = name;

        }


        abstract void display();

    }


    static class Product extends ProductFn {


        Product(int price, int quantity, String name) {

            super(price, quantity, name);

        }


        void display() {

            System.out.println("Product: " + this.name + " Price: " + this.price + " Quantity: " +
this.quantity);

        }


    }


    static class NotPaidFullAmoutException extends Exception {

        public NotPaidFullAmoutException(String s) {

            super(s);

        }

    }
```

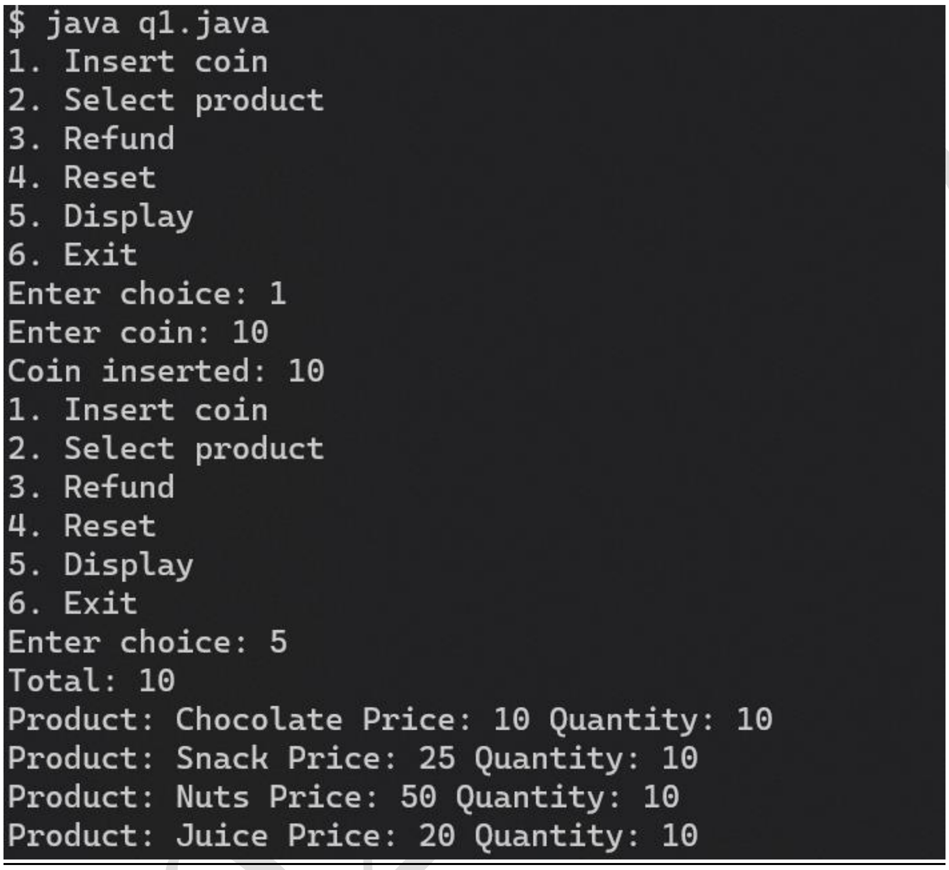
```
static class NoSufficientChangeException extends Exception {
    public NoSufficientChangeException(String s) {
        super(s);
    }
}

static class SoldOutException extends Exception {
    public SoldOutException(String s) {
        super(s);
    }
}

interface VendingMachineFn {

    void selectProduct(int choice);

    void insertCoin(int coin);

    void refund();

    void reset();

}

static class VendingMachine implements VendingMachineFn {
```

```
int total;

int[] coins = { 1, 5, 10, 25, 50 };

int[] coinCount = { 0, 0, 0, 0, 0 };


Product[] products = new Product[4];


VendingMachine() {

   products[0] = new Product(10, 10, "Chocolate");

   products[1] = new Product(25, 10, "Snack");

   products[2] = new Product(50, 10, "Nuts");

   products[3] = new Product(20, 10, "Juice");

}


public void insertCoin(int coin) {

   for (int i = 0; i < coins.length; i++) {

      if (coins[i] == coin) {

         coinCount[i]++;

         total += coin;

         System.out.println("Coin inserted: " + coin);

         return;

      }

   }

   System.out.println("Invalid coin");

}


boolean isChangeAvailable(int change) {
    int[] temp = coinCount.clone();
    for (int i = coins.length - 1; i >= 0; i--) {
        while (change >= coins[i] && temp[i] > 0) {
            change -= coins[i];
            temp[i]--;
        }
    }
    if (change == 0) {
        return true;
    }
    return false;
}

public void selectProduct(int choice) {
    try {
        if (total < products[choice].price) {
            throw new NotPaidFullAmoutException("Not paid full amount");
        }
        if (products[choice].quantity == 0) {
            throw new SoldOutException("Sold out");
        }
        if (total > products[choice].price) {
            int change = total - products[choice].price;
            if (!isChangeAvailable(change)) {
                throw new NoSufficientChangeException("No sufficient change");
            }
            System.out.println("Product: " + products[choice].name + " Change: " + change);
            products[choice].quantity--;
            total = 0;
            return;
        }
        System.out.println("Product: " + products[choice].name);
        products[choice].quantity--;
        total = 0;
    } catch (NotPaidFullAmoutException e) {
        System.out.println(e.getMessage());
    } catch (SoldOutException e) {
        System.out.println(e.getMessage());
    } catch (NoSufficientChangeException e) {
        System.out.println(e.getMessage());
    }
}

public void refund() {
    System.out.println("Refund: " + total);
    total = 0;
}

public void reset() {
    for (int i = 0; i < products.length; i++) {
        products[i].quantity = 10;
    }
```

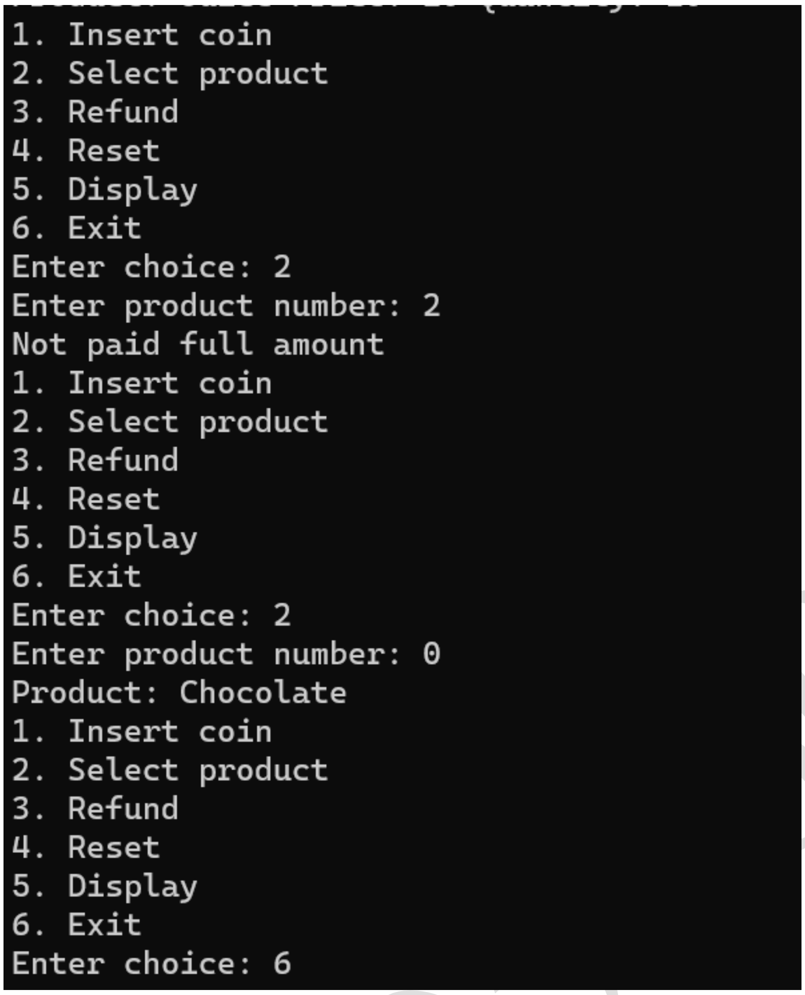
```
        for (int i = 0; i < coinCount.length; i++) {
            coinCount[i] = 0;
        }
        total = 0;
        System.out.println("Reset successful");
    }

    void display() {
        System.out.println("Total: " + total);
        for (int i = 0; i < products.length; i++) {
            products[i].display();
        }
    }

}

public static void main(String[] args) {
    VendingMachine vm = new VendingMachine();
    Scanner sc = new Scanner(System.in);
    while (true) {
        System.out.println("1. Insert coin");
        System.out.println("2. Select product");
        System.out.println("3. Refund");
        System.out.println("4. Reset");
        System.out.println("5. Display");
        System.out.println("6. Exit");
System.out.print("Enter choice: ");
int choice = sc.nextInt();
switch (choice) {
    case 1:
        System.out.print("Enter coin: ");
        int coin = sc.nextInt();
        vm.insertCoin(coin);
        break;
    case 2:
        System.out.print("Enter product number: ");
        int product = sc.nextInt();
        vm.selectProduct(product);
        break;
    case 3:
        vm.refund();
        break;
    case 4:
        vm.reset();
        break;
    case 5:
        vm.display();
        break;
    case 6:
        System.exit(0);
    default:
        System.out.println("Invalid choice");
```

```
            }
        }
    }
}
```

Output:



```
$ java q1.java
1. Insert coin
2. Select product
3. Refund
4. Reset
5. Display
6. Exit
Enter choice: 1
Enter coin: 10
Coin inserted: 10
1. Insert coin
2. Select product
3. Refund
4. Reset
5. Display
6. Exit
Enter choice: 5
Total: 10
Product: Chocolate Price: 10 Quantity: 10
Product: Snack Price: 25 Quantity: 10
Product: Nuts Price: 50 Quantity: 10
Product: Juice Price: 20 Quantity: 10
```



```
1. Insert coin
2. Select product
3. Refund
4. Reset
5. Display
6. Exit
Enter choice: 2
Enter product number: 2
Not paid full amount
1. Insert coin
2. Select product
3. Refund
4. Reset
5. Display
6. Exit
Enter choice: 2
Enter product number: 0
Product: Chocolate
1. Insert coin
2. Select product
3. Refund
4. Reset
5. Display
6. Exit
Enter choice: 6
```

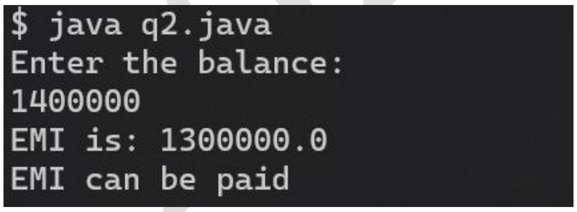
2. Calculate EMI for personal loan of Rs100000/- with the rate of interest:13% for a total of 3 years, if there is no balance in the account to pay an EMI raise a custom exception also use try catch finally mechanism

Source Code:

```
import java.util.*;


public class q2 {

    static class EMI {

        int amount;

        int rate;

        int time;

        double emi;
    EMI(int amount, int rate, int time) {
        this.amount = amount;
        this.rate = rate;
        this.time = time;
    }

    void calculate() {
        double emi;
        int n;
        n = 12 * time;
        emi = (amount * rate * Math.pow(1 + rate, n)) / (Math.pow(1 + rate, n) - 1);
        System.out.println("EMI is: " + emi);
    }
}

static class NoBalanceException extends Exception {
    public NoBalanceException(String s) {
        super(s);
    }
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    EMI e = new EMI(100000, 13, 3);

    System.out.println("Enter the balance: ");
```

```
        int balance = sc.nextInt();

        e.calculate();


        if (balance < (e.emi * 3 * 12)) {

            try {

                throw new NoBalanceException("Balance is not enough to pay the EMI for 3
years");

            } catch (NoBalanceException nbe) {

                System.out.println(nbe.getMessage());

            }

        } else {

            System.out.println("EMI can be paid");

        }


    }

}
```

<u>Output:</u>



```
$ java q2.java
Enter the balance:
1400000
EMI is: 1300000.0
EMI can be paid
```

<u>Result:</u>

Thus, exception handling has been implemented successfully.

Exp. No: 8	**Input and Output Manipulation on Files, Serialization**
Date: 25.9.22	

Aim:

To implement input and output manipulation on files and serialization

Program:

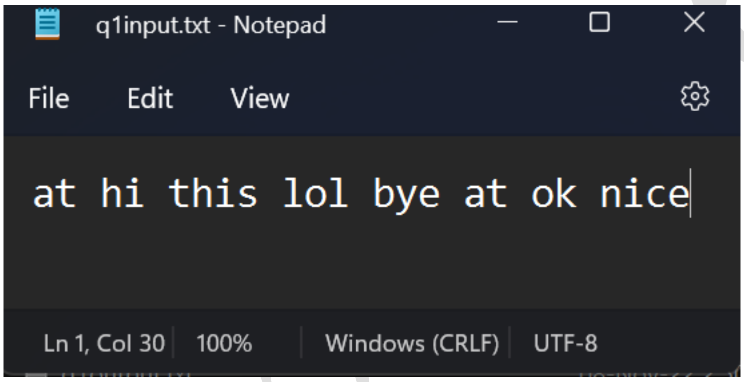
1. Write a java program using byte streams to read a text file and makes an alphabetical list of all the words in that file. Those list of words is written to another file. Improve the program so that it also keeps track of the number of times that each word occurs in the file. Two lists should be displayed in the output file. The first list contains the words in alphabetical order. The number of times that the word occurred in the file should be listed along with the word. Then write a second list to the output file in which the words are sorted according to the number of times that they occurred in the files. The word that occurred most often should be listed first.

Source Code:

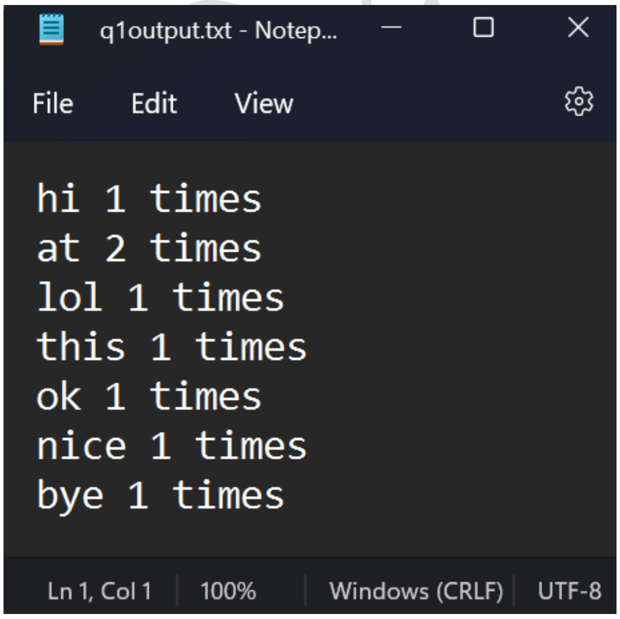
```
package exp8;


import java.util.*;

import java.io.*;

import java.util.*;


public class q1 {


    public static void main(String[] args) {


        ArrayList<String> words = new ArrayList<String>();


        try {

            FileInputStream fis = new FileInputStream("q1input.txt");
    Scanner sc = new Scanner(fis);
    while (sc.hasNext()) {
        words.add(sc.next());
    }
    sc.close();
} catch (Exception e) {
    System.out.println(e);
}

// hashmap to store words frequency
HashMap<String, Integer> hm = new HashMap<String, Integer>();

Collections.sort(words);

// add words from arraylist
for (String word : words) {
    Integer count = hm.get(word);
    hm.put(word, (count == null) ? 1 : count + 1);
}

// print it in a file
try {
    FileWriter fw = new FileWriter("q1output.txt");
    for (Map.Entry<String, Integer> entry : hm.entrySet()) {
        fw.write(entry.getKey() + " " + entry.getValue() + " times ");
        fw.write(System.getProperty("line.separator"));
    }
    fw.close();
} catch (Exception e) {
    System.out.println(e);
}

try {
    FileOutputStream fos = new FileOutputStream("q1output2.txt");
    // sort words in hashmap
    TreeMap<String, Integer> sorted = new TreeMap<String, Integer>(hm);
    for (Map.Entry<String, Integer> entry : sorted.entrySet()) {
        String str = entry.getKey() + " " + entry.getValue() + " times ";
        fos.write(str.getBytes());
        fos.write(System.getProperty("line.separator").getBytes());
    }

    // sort words according to frequency
    Collections.sort(words, new Comparator<String>() {
        @Override
        public int compare(String s1, String s2) {
            return hm.get(s2).compareTo(hm.get(s1));
        }
    });
    fos.write(System.getProperty("line.separator").getBytes());
    for (String word : words) {
        fos.write(word.getBytes());
```

```
                fos.write(" ".getBytes());
            }


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


    }
}
```

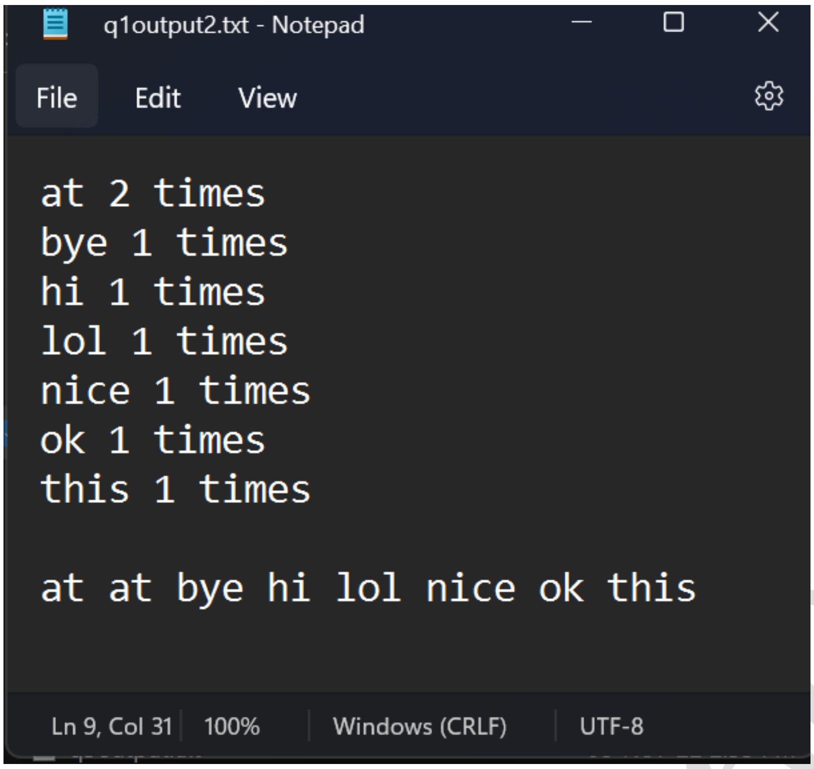
Output:



```
q1input.txt - Notepad
File  Edit  View

at hi this lol bye at ok nice

Ln 1, Col 30 | 100% | Windows (CRLF) | UTF-8
```



```
q1output.txt - Notep...
File  Edit  View

hi 1 times
at 2 times
lol 1 times
this 1 times
ok 1 times
nice 1 times
bye 1 times

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8
```



```
at 2 times
bye 1 times
hi 1 times
lol 1 times
nice 1 times
ok 1 times
this 1 times

at at bye hi lol nice ok this
```

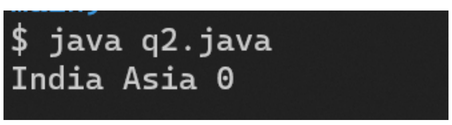
2. Write a java program to have country class and serialize country name and continent name to which the country belongs to. You don’t want to serialize population attribute as it will change with time, so declare population as transient. Perform deserialization, Print country name, continent name and population of the country.

Source Code:

```
package exp8;

import java.io.*;


public class q2 {


    public static class country implements java.io.Serializable {

        String name;

        String continent;

        transient int population;


        public country(String name, String continent, int population) {
```

```
        this.name = name;

        this.continent = continent;

        this.population = population;

    }

}


public static void main(String[] args) {

    // serialize name and continent

    try {

        FileOutputStream fos = new FileOutputStream("q2output.txt");

        ObjectOutputStream oos = new ObjectOutputStream(fos);

        oos.writeObject(new country("India", "Asia", 1000000000));

        oos.close();

        fos.close();

    } catch (Exception e) {

        System.out.println(e);

    }


    // deserialize name and continent

    try {

        FileInputStream fis = new FileInputStream("q2output.txt");

        ObjectInputStream ois = new ObjectInputStream(fis);

        country c = (country) ois.readObject();

        System.out.println(c.name + " " + c.continent + " " + c.population);

        ois.close();

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

    }

}
```

Output:



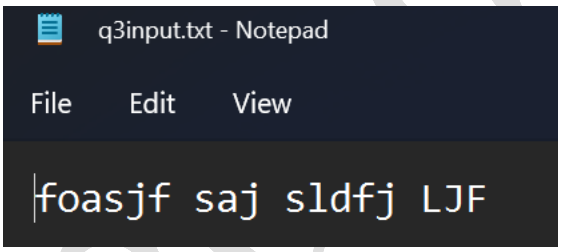
```
$ java q2.java
India Asia 0
```

3. Write a Java program using character stream classes that copies one file content to another, replacing all lower characters by their upper case equivalents.

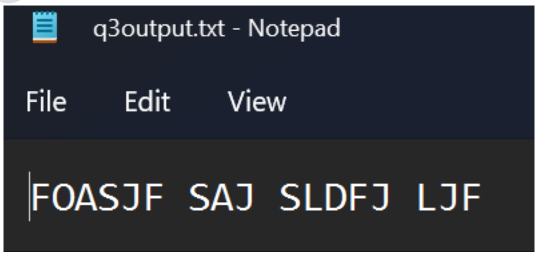
Source Code

```
package exp8;

import java.io.*;

public class q3 {

    public static void main(String[] args) {

        try {
            FileReader fr = new FileReader("q3input.txt");
            FileWriter fw = new FileWriter("q3output.txt");
            int c;
            while ((c = fr.read()) != -1) {
                if (Character.isLowerCase(c)) {
                    c = Character.toUpperCase(c);
                }
                fw.write(c);
            }
            fr.close();
            fw.close();
        } catch (Exception e) {
            System.out.println(e);
        }

    }

}
```

Output:


q3input.txt - Notepad

```
foasjf saj sldfj LJF
```


q3output.txt - Notepad

```
FOASJF SAJ SLDFJ LJF
```

Result:

Thus, input and output manipulation on files and serialization has been implemented successfully.

Exp. No: 9	Client Server network app using java sockets
Date: 12.10.22	

Aim:

To implement client server network app using java sockets.

Program:

1. Write a java program to simulate chatbot application which does two way communication between client and server using connection oriented protocol.

Source Code:

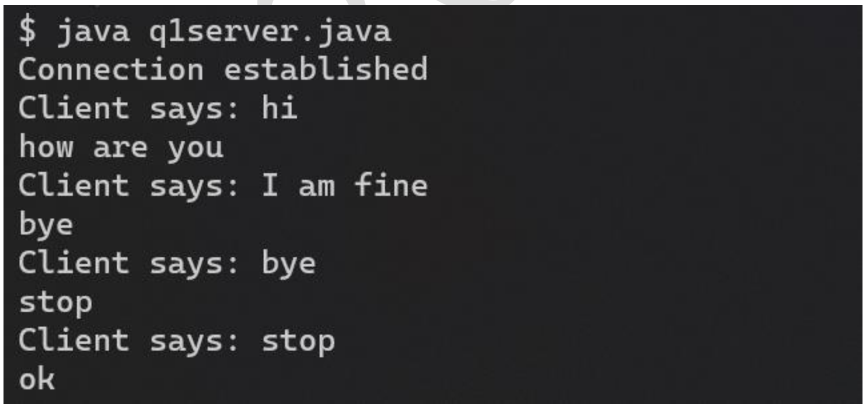
Q1Client

```
import java.io.*;

import java.net.*;


public class q1client {

    public static void main(String[] args) {

        try {

            Socket s = new Socket("localhost", 1234);

            DataInputStream dis = new DataInputStream(s.getInputStream());

            DataOutputStream dos = new DataOutputStream(s.getOutputStream());

            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            String str = "", str2 = "";

            while (!str.equals("stop")) {

                str = br.readLine();

                dos.writeUTF(str);

                dos.flush();

                str2 = dis.readUTF();

                System.out.println("Server says: " + str2);
            }
            dos.close();
            s.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }

}
```

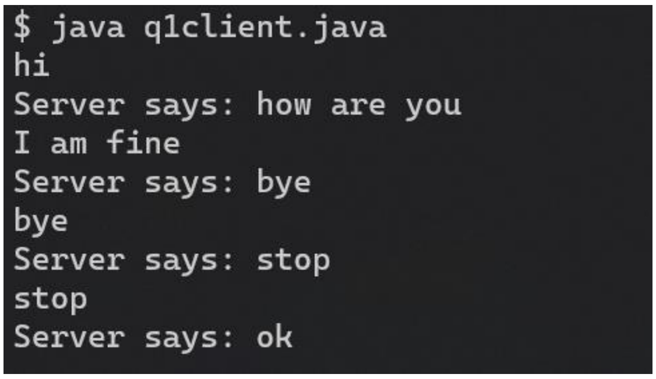
Q1Server

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

public class q1server {

    public static void main(String[] args) {

        // tcp server
        try {
            ServerSocket ss = new ServerSocket(1234);
            Socket s = ss.accept();
            System.out.println("Connection established");
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new DataOutputStream(s.getOutputStream());
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            String str = "", str2 = "";
            while (!str.equals("stop")) {
                str = dis.readUTF();
                System.out.println("Client says: " + str);
                str2 = br.readLine();
                dos.writeUTF(str2);
                dos.flush();
            }
            dis.close();
            s.close();
            ss.close();
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```

Output:



```
$ java q1server.java
Connection established
Client says: hi
how are you
Client says: I am fine
bye
Client says: bye
stop
Client says: stop
ok
```



```
$ java q1client.java
hi
Server says: how are you
I am fine
Server says: bye
bye
Server says: stop
stop
Server says: ok
```

2. Write a java program to perform two way communication between client and server where message sent by client are converted to upper case and displayed in server side, message sent by server should be converted to lower case and displayed in client side. Implement the scenario using connectionless protocol.

<u>Source Code:</u>

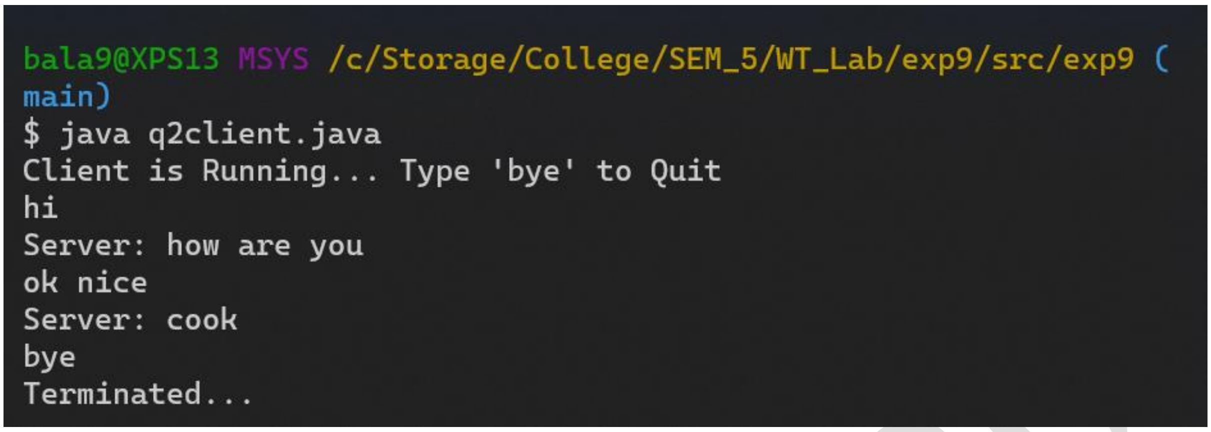
Q2Client

```
import java.io.*;

import java.net.*;


class UDPCHATClient {

    public static DatagramSocket clientsocket;

    public static DatagramPacket dp;

    public static BufferedReader dis;

    public static InetAddress ia;

    public static byte buf[] = new byte[1024];

    public static int cport = 789, sport = 790;


    public static void main(String[] a) throws IOException {

        clientsocket = new DatagramSocket(cport);

        dp = new DatagramPacket(buf, buf.length);

        dis = new BufferedReader(new InputStreamReader(System.in));

        ia = InetAddress.getLocalHost();
        System.out.println("Client is Running... Type 'bye' to Quit");
        while (true) {
            String str = new String(dis.readLine());
            // convert to upper case
            str = str.toUpperCase();
            buf = str.getBytes();
            if (str.equals("BYE")) {
                System.out.println("Terminated...");
                clientsocket.send(new DatagramPacket(buf, str.length(), ia, sport));
                break;
            }
            clientsocket.send(new DatagramPacket(buf, str.length(), ia, sport));
            clientsocket.receive(dp);
            String str2 = new String(dp.getData(), 0, dp.getLength());
            System.out.println("Server: " + str2);
        }
    }
}
```

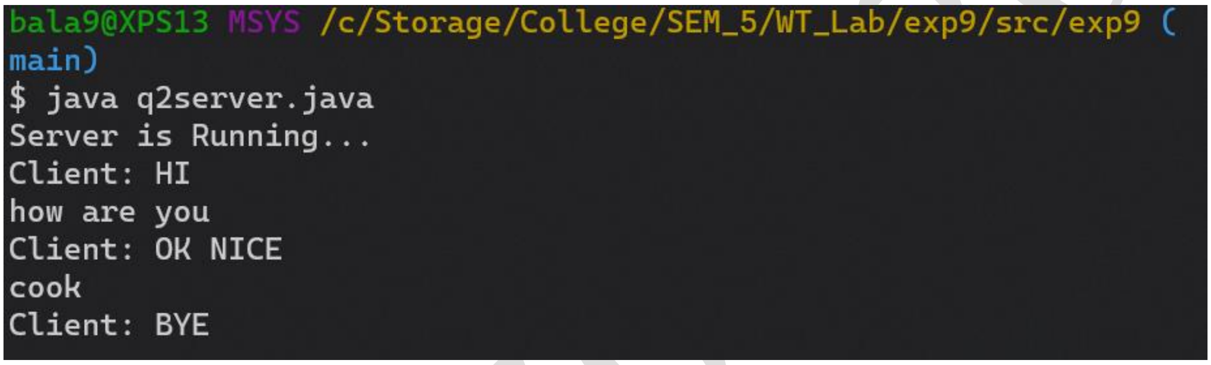
Q2Server

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

class UDPCHATServer {
    public static DatagramSocket serversocket;
    public static DatagramPacket dp;
    public static BufferedReader dis;
    public static InetAddress ia;

    public static byte buf[] = new byte[1024];

    public static int cport = 789, sport = 790;


    public static void main(String[] a) throws IOException {

        serversocket = new DatagramSocket(sport);

        dp = new DatagramPacket(buf, buf.length);

        dis = new BufferedReader(new InputStreamReader(System.in));

        ia = InetAddress.getLocalHost();

        System.out.println("Server is Running...");

        while (true) {

            serversocket.receive(dp);

            String str = new String(dp.getData(), 0, dp.getLength());

            if (str.equals("bye")) {

                System.out.println("Terminated...");

                break;

            }

            System.out.println("Client: " + str);

            String str1 = new String(dis.readLine());

            // convert to lower case

            str1 = str1.toLowerCase();

            buf = str1.getBytes();

            serversocket.send(new DatagramPacket(buf, str1.length(), ia, cport));

        }

    }

}
```

Output:



```
bala9@XPS13 MSYS /c/Storage/College/SEM_5/WT_Lab/exp9/src/exp9 (
main)
$ java q2client.java
Client is Running... Type 'bye' to Quit
hi
Server: how are you
ok nice
Server: cook
bye
Terminated...
```



```
bala9@XPS13 MSYS /c/Storage/College/SEM_5/WT_Lab/exp9/src/exp9 (
main)
$ java q2server.java
Server is Running...
Client: HI
how are you
Client: OK NICE
cook
Client: BYE
```

Result:

Thus, networking has been implemented successfully in java.

Exp. No: 10	**Multithreading**
Date: 26.10.22	

Aim:

To implement multithreading in java

Program:

1. Write a Java program that creates three threads. First thread displays “Hello!” every one second, the second thread displays “Wear Mask !” every two seconds and “Use Sanitizer!” every 5 seconds

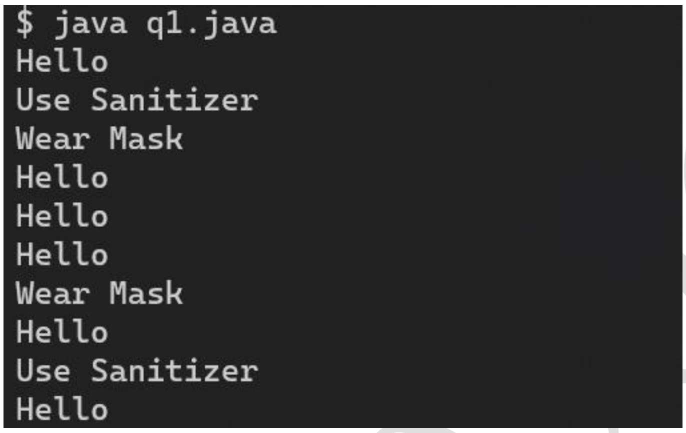
Source Code:

```
public class q1 {

   public static void main(String[] args) {



       Thread t1 = new Thread(new Runnable() {

           @Override

           public void run() {

               while (true) {

                   System.out.println("Hello");

                   try {

                       Thread.sleep(1000);

                   } catch (InterruptedException e) {

                       e.printStackTrace();

                   }

               }

           }

       });
```

```
Thread t2 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Wear Mask");
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});

Thread t3 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Use Sanitizer");
            try {
                Thread.sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
```

```
        });


        t1.start();

        t2.start();

        t3.start();

    }

}
```

Output:



```
$ java q1.java
Hello
Use Sanitizer
Wear Mask
Hello
Hello
Hello
Wear Mask
Hello
Use Sanitizer
Hello
```

2. Write a Java program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which thread is long lasting

Source Code:

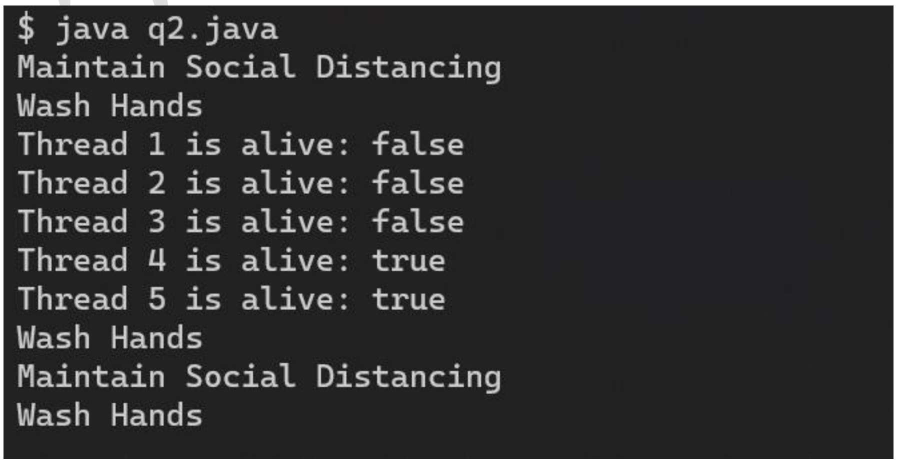
```
import java.lang.Thread;


public class q2 {

    public static void main(String[] args) {

        Thread t1 = new Thread(new Runnable() {

            @Override

            public void run() {

                while (true) {
```

```
            System.out.println("Hello");
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});

Thread t2 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Wear Mask");
            try {
                Thread.sleep(3000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});

Thread t3 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Use Sanitizer");
            try {
                Thread.sleep(5000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});

Thread t4 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Wash Hands");
            try {
                Thread.sleep(7000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});
```

```
Thread t5 = new Thread(new Runnable() {
    @Override
    public void run() {
        while (true) {
            System.out.println("Maintain Social Distancing");
            try {
                Thread.sleep(9000);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
});

t1.setPriority(1);
t2.setPriority(2);
t3.setPriority(3);
t4.setPriority(4);
t5.setPriority(5);

t4.start();
t5.start();

try {
    t5.sleep(1000);
} catch (InterruptedException e) {
```

```
            e.printStackTrace();
        }

        try {
            t4.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        // check if thread is alive
        System.out.println("Thread 1 is alive: " + t1.isAlive());
        System.out.println("Thread 2 is alive: " + t2.isAlive());
        System.out.println("Thread 3 is alive: " + t3.isAlive());
        System.out.println("Thread 4 is alive: " + t4.isAlive());
        System.out.println("Thread 5 is alive: " + t5.isAlive());

    }
}
```

Output:



```
$ java q2.java
Maintain Social Distancing
Wash Hands
Thread 1 is alive: false
Thread 2 is alive: false
Thread 3 is alive: false
Thread 4 is alive: true
Thread 5 is alive: true
Wash Hands
Maintain Social Distancing
Wash Hands
```

3. Write a java program that implements a multi-thread applications that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the thread will print the value of the number

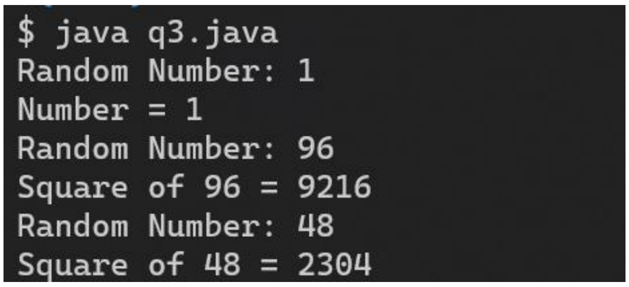
Source Code

```
public class q3 {


    static class SquareThread extends Thread {
        int number;


        SquareThread(int randomNumbern) {
            number = randomNumbern;
        }


        public void run() {
            System.out.println("Square of " + number + " = " + (number * number));
        }
    }


    static class PrintThread extends Thread {
        int number;


        PrintThread(int randomNumber) {
            number = randomNumber;
        }


        public void run() {
```

```
        System.out.println("Number = " + number);
    }
}

public static void main(String[] args) {

    Thread t1 = new Thread(new Runnable() {
        @Override
        public void run() {
            while (true) {
                int randomNumber = (int) (Math.random() * 100);
                System.out.println("Random Number: " + randomNumber);
                if (randomNumber % 2 == 0) {
                    new SquareThread(randomNumber).start();
                } else {
                    new PrintThread(randomNumber).start();
                }
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    });
```

```
        t1.start()
    }
}
```

Output:



```
$ java q3.java
Random Number: 1
Number = 1
Random Number: 96
Square of 96 = 9216
Random Number: 48
Square of 48 = 2304
```

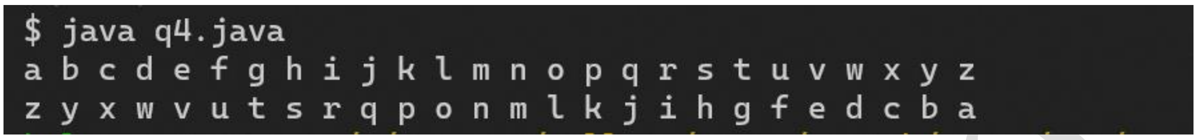
4. Write a java program that implements a multi-thread applications that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the thread will print the value of the number

Source Code

```
public class q4 {
    public static void main(String[] args) {
        Thread t1 = new Thread(new Runnable() {
            @Override
            public void run() {
                for (char i = 'a'; i <= 'z'; i++) {
                    System.out.println(i);
                    try {
                        Thread.sleep(100);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            }
    });
    Thread t2 = new Thread(new Runnable() {
        @Override
        public void run() {
            for (char i = 'z'; i >= 'a'; i--) {
                System.out.println(i);
                try {
                    Thread.sleep(200);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    });
    t1.start();
    try {
        t1.join();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    t2.start();
    try {
        t2.join();
    } catch (InterruptedException e) {
        e.printStackTrace( )
}
```

```
    }
}
```

Output:



```
$ java q4.java
a b c d e f g h i j k l m n o p q r s t u v w x y z
z y x w v u t s r q p o n m l k j i h g f e d c b a
```

Result:

Thus, multithreading has been implemented successfully.

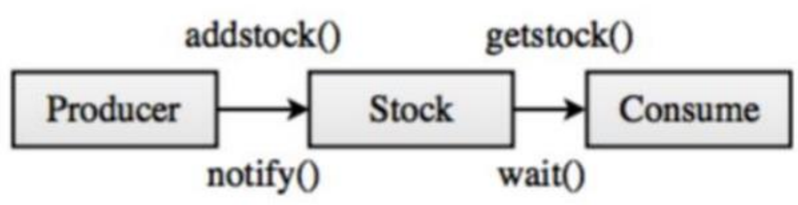
Exp. No: 11	Inter Thread Communication
Date: 29.10.22	

Aim:

To implement inter thread communication

Program:

1. Write a Java program that implements inter thread communication for the following scenario



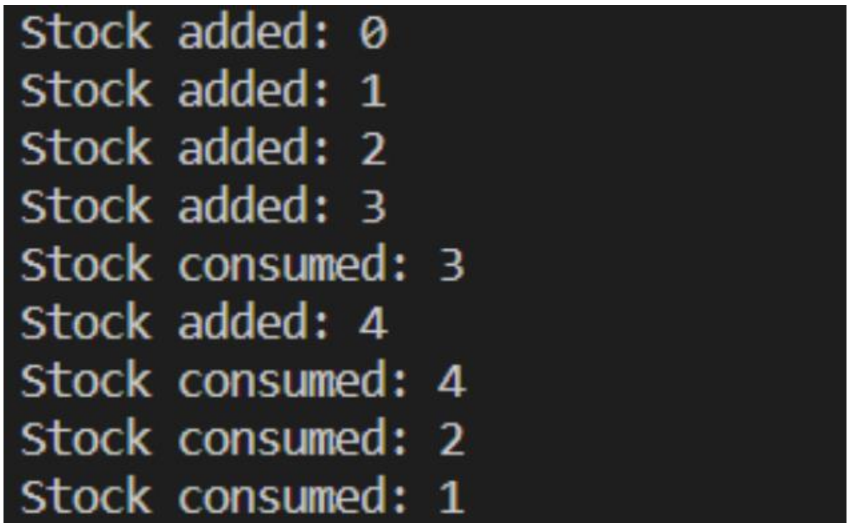
Source Code:

```
import java.util.*;

class Thread1 extends Thread{

 PC obj;

 Thread1(PC obj){

 this.obj= obj;

 }

 public void run(){

 try{

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

 obj.addstock(i);

 }catch(InterruptedException e){

 e.printStackTrace();

 }

 }
```

```
}
class Thread2 extends Thread{
 PC obj;
 Thread2(PC obj){
 this.obj= obj;
 }
 public void run(){
 try{
 while(obj.st.size()!= 0)
 obj.getstock();
 }catch(InterruptedException e){
 e.printStackTrace();
 }
 }
}
public class ProducerConsumer {
 public static void main(String[] args) throws InterruptedException{
 final PC pc= new PC();
 Thread1 t1= new Thread1(pc);
 Thread2 t2= new Thread2(pc);

 t1.start();
 t2.start();
 t1.join();
 t2.join();
 }
```

```
}
class PC{
 Stack<Integer> st= new Stack<Integer>();
 public void addstock(int num) throws InterruptedException{
 synchronized(this){
 if(st.size()>3) wait();
 st.push(num);
 System.out.println("Stock added: "+ num);
 }
 }
 public void getstock() throws InterruptedException{
 Thread.sleep(1000);
 if(st.size()== 0) wait();
 Scanner s= new Scanner(System.in);
 synchronized(this){
 System.out.println("Stock consumed: "+ st.pop());
 notify();
 //Thread.sleep(2000);
 }
 }
}
```

Output:



```
Stock added: 0
Stock added: 1
Stock added: 2
Stock added: 3
Stock consumed: 3
Stock added: 4
Stock consumed: 4
Stock consumed: 2
Stock consumed: 1
```

Result:

Thus, inter thread communication has been implemented successfully.