1. Write a program in Java to make such a pattern like right angle triangle with number increased by 1. The pattern like:

```
1

2 3

4 5 6

7 8 9 10
```

Number of rows will be entered by user.

Ans:

MainClass:

```
import java.util.Scanner;

public class MainClass {

    public static void main(String[] args) {

        RightAngleTriangle r= new RightAngleTriangle();

        System.out.print("Enter the Number of rows : ");

        int no=new Scanner(System.in).nextInt();

        r.displayPattern(no);

    }}
```

RightAngleTriangle :

```
public class RightAngleTriangle {

    public void displayPattern(int no) {

        int count = 1;

        for (int i = 1; i <= no; i++) {

            for (int j = 1; j <= i; j++) {

                System.out.print(count++ + "\t");
```

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

2. Write a Java program to display Pascal's triangle.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

Number of rows will be entered by user.

Ans:

MainClass:

```
import java.util.Scanner;
public class MainClass {
        public static void main(String[] args) {
                PascalTriangle p = new PascalTriangle();
                System.out.print("Enter the Number of rows : ");
                int no= new Scanner(System.in).nextInt();
                p.displayPattern(no);
        }}
```

PascalTriangle :

```
public class PascalTriangle {
        public void displayPattern(int no) {
```

```
        int num;

        for (int i = 0; i < no; i++) {

            for (int j = i; j < no; j++) {

                System.out.print(" ");

            }num = 1;

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

                System.out.print(num + " ");

                num = num * (i - j) / (j + 1);

            }

            System.out.println();}

    }}
```

3. Write a Java program to count the letters, spaces, numbers, and other characters of an input string.

Ans:

StringDemo :

```
public class StringDemo {

    public int countLetter(String s) {

        int letterCount = 0;

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

            if (Character.isAlphabetic(s.charAt(i))) {

                letterCount++;

            }}

        return letterCount;}
    public int countSpaces(String s) {
        int spaceCount=0;
        for (int i = 0; i < s.length(); i++) {
            if (Character.isWhitespace(s.charAt(i))) {
                spaceCount++;
            }
        }
        return spaceCount;}
    public int countNumbers(String s) {
        int numCount=0;
        for (int i = 0; i < s.length(); i++) {
            if (Character.isDigit(s.charAt(i))) {
                numCount++;
            }}
        return numCount;}
    public int countSpecialChar(String s) {
        int specialCharCount=0;
        for (int i = 0; i < s.length(); i++) {
            if
(!(Character.isLetterOrDigit(s.charAt(i))||Character.isWhitespace(s.charAt(i)))) {
                specialCharCount++;
            }}
    return specialCharCount;
    }}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        StringDemo sd = new StringDemo();
        String s = "Hello!! I am Java Developer, I am working at Proventeq from 12 May 2022";
        System.out.println("Total Letters in Given String = "+sd.countLetter(s));
        System.out.println("Total Spaces in Given String = "+sd.countSpaces(s));
        System.out.println("Total Numbers in Given String = "+sd.countNumbers(s));
        System.out.println("Total Special Character in Given String = "+sd.countSpecialChar(s));
    }}
```

4. Write a Java program to compute the sum of the first 100 prime numbers.

Ans:

MainClass:

```
public class MainClass {
    public static void main(String[] args) {
        PrimeNumber p = new PrimeNumber();
        System.out.println("Sum of 1st 100 Prime Num = "+p.sumOfPrimeNum());
    }}
```

PrimeNumber :

```
public class PrimeNumber {
    public boolean isPrime(int num) {
        boolean b = false;
        if (num == 1) {
            b = true;}
        for (int i = 2; i <= num / 2; i++) {
            if (num % i == 0)
                b = true;}
        return !b;}
    public int sumOfPrimeNum() {
        int count = 0, sum = 0;
        for (int i = 1; i <= 100000; i++) {
            if (count != 100) {
                if (isPrime(i)) {
                    sum += i;
                    count++;
                }}}
        return sum;
}}
```

5. Write a Java program to break an integer into a sequence of individual digits.

Input six non-negative digits: 123456

ExpectedOutput: 1 2 3 4 5 6

Ans:

MainClass:

```
public class MainClass {
    public static void main(String[] args) {
        BreakInteger b = new BreakInteger();
        int a[] = b.breakNum(123456);
```

```
                for(int no:a) {
                        System.out.print(no+" ");
                }}}
```

BreakInteger :

```
public class BreakInteger {
        public int[] breakNum(int no) {
                int length= new Integer(no).toString().length();
                int a[]=new int[length];
                while(no!=0) {
                        for(int i=a.length-1; i>=0;i--) {
                                a[i]=no%10;
                                no/=10;
                        }}
                return a;
        }}
```

6. W.A.J.P to insert an element (specific position) into an array.

Ans:

getArray:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Scanner;
public class ArrayDemo {
ArrayList<Integer> al = new ArrayList();
public void getArray() {
        System.out.print("Enter the Array length : ");
        int len = new Scanner(System.in).nextInt();
        System.out.println("Enter the elements : ");
        int a[] = new int[len];
        for (int i = 0; i < a.length; i++) {
            int no = new Scanner(System.in).nextInt();
            a[i] = no;
}
        System.out.println("Given array : " + Arrays.toString(a));
        for (int i : a) {
            al.add(i);
        }
        System.out.println("--------------------------------------------------");
        System.out.print("Enter the position where you want to store element : ");
        int pos = new Scanner(System.in).nextInt();
        System.out.print("Enter the new element : ");
        int ele = new Scanner(System.in).nextInt();
        System.out.println("--------------------------------------------------");
        insertEleInArray(a, pos, ele);
    }
public void insertEleInArray(int a[], int ind, int ele) {
        int b[] = new int[a.length + 1];
        al.add(ind, ele);
        for (int i = 0; i < al.size(); i++) {
            b[i] = al.get(i);
        }
        System.out.println("After array : " + Arrays.toString(b));
    }}
```

MainClass:

```
public class MainClass {
    public static void main(String[] args) {
        ArrayDemo ad = new ArrayDemo();
        ad.getArray();
    }}
```

7. W.A.J.P to find the common elements between two arrays (string values)

Ans:

CommonElementsArray

```
import java.util.ArrayList;
import java.util.Scanner;
public class CommonElementsArray {
    ArrayList<String> hs1 = new ArrayList();
    ArrayList<String> hs2 = new ArrayList();
    String s1[] = null;
    String s2[] = null;
public void getArrays() {
        System.out.print("Enter size of First Array : ");
        int size = new Scanner(System.in).nextInt();
        String s1[] = new String[size];
```

```
            for (int i = 0; i < size; i++) {
                String str = new Scanner(System.in).nextLine();
                s1[i] = str;
            }
System.out.print("Enter size of Second Array : ");
            int size1 = new Scanner(System.in).nextInt();
            String s2[] = new String[size1];
            for (int i = 0; i < size1; i++) {
                String str = new Scanner(System.in).nextLine();
                s2[i] = str;
            }
            printArray(s1,s2);
      }
public void printArray(String s1[], String s2[]) {
            System.out.print("\nFirst Array is : ");
            for (String s : s1) {
                System.out.print(s + " ");
                hs1.add(s);
            }
System.out.print("\nSecond Array is : ");
            for (String s : s2) {
                System.out.print(s + " ");
                hs2.add(s);
            }
```

```
    }
public void getCommonElements() {
        hs1.retainAll(hs2);
        System.out.println("\n---------------------------------");
        System.out.print("Common Elements : ");
        for (String s : hs1) {
            System.out.print(s + " ");
        }
        System.out.println("\n---------------------------------");
    }}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        CommonElementsArray ce = new CommonElementsArray();
        ce.getArrays();
        ce.getCommonElements();
    }}
```

8. W.A.J.P to segregate all 0s on left side and all 1s on right side of a given array of 0s and 1s.

Original Array: [0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1]

Array after segregation is: [0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1]

Ans:

SegregateArray :

```
import java.util.Arrays;
public class SegregateArray {
	public void segregateArray(int a[]) {

		System.out.println("Original given Array is : " + Arrays.toString(a));

		Arrays.sort(a);

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

		System.out.print("Array after segregation : " + Arrays.toString(a));

	}}
```

MainClass :

```
public class MainClass {

	public static void main(String[] args) {

		SegregateArray sg = new SegregateArray();

		int a[] = { 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1 };

		sg.segregateArray(a);

	}}
```

9. W.A.J.P to cyclically rotate a given array clockwise by one.

Original array: [10, 20, 30, 40, 50, 60]

Rotated array: [60, 10, 20, 30, 40, 50]

Ans:

RotateArray :

```
public class RotateArray {

	public int[] rotateAray(int a[]) {

		System.out.println("Original Array ");

		for (int b : a) {

			System.out.print(b + " ");

		}
```

```
        int b[] = new int[a.length];

        b[0] = a[a.length - 1];

        for (int i = 0; i < a.length - 1; i++) {

            b[i + 1] = a[i];

        }

        return b;

    }

public void printArray(int a[]) {

        int b[] = rotateAray(a);

        System.out.println("\nAfter array");

        for (int c : b) {

            System.out.print(c + " ");

        }

    }}
```

MainClass :

```
public class MainClass {

    public static void main(String[] args) {

        RotateArray r = new RotateArray();

        r.printArray(new int[] { 10, 20, 30, 40, 50, 60 });

    }}
```

10. W.A.J.P to find the second most frequent character in a given string.

The given string is: successes

The second most frequent char in the string is: c

Ans:

SecondFrequent :

```
import java.util.Arrays;

import java.util.HashMap;

import java.util.Map.Entry;

public class SecondFrequent {

	public void secondFrequentChar(String s) {

		s = s.toLowerCase();

		HashMap<Character, Integer> m = new HashMap();

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

			int count = 1;

			for (int j = 0; j < s.length(); j++) {

				if (i != j && s.charAt(i) == s.charAt(j)) {

					count++;

				}

				m.put(s.charAt(i), count);

			}	}

		System.out.println(m);

		Object[] repeate = new Object[m.size()];

		System.out.println(m);

		repeate = m.values().toArray();

		Arrays.sort(repeate);

		System.out.println(Arrays.toString(repeate));

		int a = (int) repeate[repeate.length - 2];

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

```
        System.out.print("Second Frequent Character is : ");
        for (Entry<Character, Integer> entry : m.entrySet()) {
            if (a == entry.getValue()) {
                System.out.print(entry.getKey() + " ");
            }
        }   }}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        SecondFrequent sf= new SecondFrequent();
        String s= "succcess";
        sf.secondFrequentChar(s);
    }}
```

11. W.A.J.P to find first non-repeating character in a string.

The given string is: gibblegabbler

The first non-repeated character in String is: i

Ans:

NonRepeatString :

```
public class NonRepeatString {
    public void nonRepeat(String s) {
        System.out.println("The given string is: " + s);
        for (int i = 0; i < s.length(); i++) {
            boolean b = false;
            for (int j = 0; j < s.length(); j++) {
                if (i != j && s.charAt(i) == s.charAt(j)) {
                    b = true;
                    break;
                }   }
            if (!b) {
                System.out.println("The first non repeated character in String is: " + s.charAt(i));
                break;
            }
        }}}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        NonRepeatString nr = new NonRepeatString();
        String s = "gibblegabbler";
        nr.nonRepeat(s);
    }}
```

12. W.A.J.P to divide a string in an equal part.

The given string is: abcdefghijklmnopqrstuvwxy

The string divided into 5 parts and they are:

abcde ,fghij ,klmno ,pqrst ,uvwxy

Ans:

StringDemo :

```
public class StringDemo {
    public void divideString(String s) {
        int len = s.length();
        int temp = 0, n = 5;
        String a[] = new String[n];
        if (len % n != 0) {
            System.out.println("String can't divided equal parts");
        } else {
            for (int i = 0; i < len; i = i + n) {
                String part = s.substring(i, i + n);
                a[temp] = part;
                temp++;
            }
            System.out.println(n + " equal parts of given String are : ");
            for (int i = 0; i < a.length; i++) {
                System.out.println(a[i]);
            }
        }}}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        StringDemo sd = new StringDemo();
        String s = "abcdefghijklmnopqrstuvwxy";
        sd.divideString(s);
    }}
```

13. Create an abstract class 'Bank' with an abstract method 'getBalance'. $100, $150 and $200 are deposited in

banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method

named 'getBalance'. Call this method by creating an object of each of the three classes.

Ans:

Bank :

```
public abstract class Bank {

        public abstract void getBalance();

}
```

BankA :

```
public class BankA extends Bank{

        @Override

        public void getBalance() {

                int bal = 100;

                System.out.println("Balnace of BankA = $" + bal);

        }}
```

BankB :

```
public class BankB extends Bank{

        @Override

        public void getBalance() {

                int bal=150;

                System.out.println("Balnace of BankB = $"+bal);

        }}
```

BankC :

```
public class BankC extends Bank{
    @Override
    public void getBalance() {
        int bal=200;
        System.out.println("Balnace of BankC = $"+bal);
    }}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        BankA a= new BankA();
        a.getBalance();
        BankB b= new BankB();
        b.getBalance();
        BankC c= new BankC();
        c.getBalance();
    }}
```

14. Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named

'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and

"This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is

a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

Ans:

Shape :

```
public class Shape {
        public void printShape() {
                System.out.println("This is Shape");}
}
```

Rectangle :

```
public class Rectangle extends Shape{
        public void printRectangle() {
                System.out.println("This is rectangular shape");
        }}
```

Circle :

```
public class Circle extends Shape {
        public void printCircle() {
                System.out.println("This is Circular Shape");
        }}
```

Square :

```
public class Square extends Rectangle {
        public void printSquare() {
                System.out.println("Square is Rectangle");
        }}
```

MainClass :

```
public class MainClass {
        public static void main(String[] args) {
                Square s = new Square();
                s.printShape();
                s.printRectangle();
        }}
```

15. W.A.J. P to demonstrate multiple catch blocks, (one is to handle divide by zero exception and another one is to

handle ArrayIndexOutOfBoundException)

int a [] =new int [5];

a [5] =30/0;

Ans:

ExceptionDemo :

```
public class ExceptionDemo {
    public void divideByZero(int a[]) {
        try {
            int result = a[1] / 0;
            } catch (ArithmeticException ae) {
            System.out.println(ae);
        }
        try {
            System.out.println(a[5]);
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println(e);
        }}}
```

MainClass:

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

        ExceptionDemo e = new ExceptionDemo();

        e.divideByZero(new int[] { 10, 20, 30 });

    } }
```

16. W.A.J.P to create the validate method that takes integer value as a parameter. If the age is less than 18, then throw an ArithmeticException otherwise print a message welcome to vote.

Ans:

AgeDemoException :

```
public class AgeDemoException {

    public void validateAge(int age) {

        if(age<18) {

            throw new ArithmeticException("not valid");

        }

        else

            System.out.println("Welcome to Vote");

    }}
```

MainClass :

```
public class MainClass {

    public static void main(String[] args) {

        AgeDemoException ad= new AgeDemoException();

        System.out.print("Enter your Age : ");

        int age= new Scanner(System.in).nextInt();

        ad.validateAge(16);

    }}
```

17. W.A.J.Pto create a custom exception if Customer withdraw amount which is greater than account balance then

program will show custom exception otherwise amount will deduct from account balance. Account balance

is:2000 Enter withdraw amount:2500 Sorry, insufficient balance, you need more 500 Rs. To perform this

transaction

Ans:

InsufficientFundException :

```
public class InsufficientFundException extends RuntimeException {

	private String msg = "Sorry, Insufficient Balance!!!";

	public InsufficientFundException() {

	}

	public InsufficientFundException(String msg) {

		this.msg = msg;

	}

	@Override

	public String toString() {

		return "InsufficientFundException : " + msg;

	}}
```

Transaction :

```
public class Transaction {

	public void withdrawlAmt(double dbAmt) {

		double acBal = 2000, diff=0;
```

```
        diff=acBal-dbAmt;
        if(diff<0)
            diff*=(-1);
        if(acBal>=dbAmt) {
            acBal-=dbAmt;
            System.out.println("Updated Bal is : "+acBal);
        }
        else
            throw new InsufficientFundException("You need more "+diff+" to perform this transaction");
    }
}
```

MainClass :

```
import java.util.Scanner;
public class MainClass {
    public static void main(String[] args) {
        Transaction t =new Transaction();
        System.out.print("Enter the Withdrawl Amount : ");
        double dbAmt = new Scanner(System.in).nextDouble();
        t.withdrawlAmt(dbAmt);
    }}
```

18. W.A.J.P to start the same Thread twice by calling start () method twice.

```
TestThreadTwice1 t1=new TestThreadTwice1();
t1.start();
```

t1.start();

Ans:

ThreadStartDemo :

```
public class ThreadStartDemo implements Runnable {

    public void displayMsg() {

        System.out.println("This is Thread");

    }

    @Override

    public void run() {

        displayMsg();

    }}
```

MainClass :

```
public class MainClass {

    public static void main(String[] args) {

        ThreadStartDemo tsd = new ThreadStartDemo();

        Thread t = new Thread(tsd);

        t.start();

        t.start();

    }}
```

19. W.A.J.P to create 2 threads and make one thread as DaemonThread by using setDaemon () method of Thread

class and check whether the thread is set daemon or not by using isDaemon () method.

TestDaemonThread2 t1=new TestDaemonThread2();

TestDaemonThread2 t2=new TestDaemonThread2();

```
t1.start();

t1.setDaemon(true);//will throw exception here

t2.start()
```

Ans:

ThreadOne :

```
public class ThreadOne extends Thread {

        public void run() {

                System.out.println("This is First Thread");

        } }
```

ThreadSecond :

```
public class ThreadSecond extends Thread {

        public void run() {

                System.out.println("This is Second Thread");

        } }
```

MainClass :

```
public class MainClass {

        public static void main(String[] args) {

                ThreadOne to = new ThreadOne();

                ThreadSecond dt = new ThreadSecond();

                Thread t1 = new Thread(to);

                Thread t2 = new Thread(dt);

                dt.setDaemon(true);

                t1.start();

                t2.start();
                System.out.println("------------------------------------");
                if (to.isDaemon()) {
                    System.out.println("Thread One is Daemon Thread");
                } else
                    System.out.println("Thread One is not Daemon Thread");
                if (dt.isDaemon()) {
                    System.out.println("Thread Second is Daemon Thread");
                } else
                    System.out.println("Thread Second is not Daemon Thread");
        } }
```

20. W.A.J.P to create one class named First having method display () which display message with “[“as,

```
public void display (String msg)
{
System.out.print("["+msg);
try {
Thread.sleep(1000);
} catch(InterruptedExceptione)
{
e. printStackTrace ();
}
System.out.println("]");
}
```

Then create another class named Second by extending First class in it and use synchronized () method and use

```
display () method in it. public void run ()

{

synchronized(fobj) //Synchronized block

{

fobj.display(msg);

}

}

First fnew= new First();

Second ss= new second(fnew, "welcome");

Second ss1= new second (fnew,"new");

Second ss2 = new second(fnew, "programmer");

o/p-

[welcome]

[new]

[programmer]
```

Ans:

First :

```
public class First {

    public void display(String msg) {

        System.out.print("[" + msg);

        try {

            Thread.sleep(1000);

        } catch (InterruptedException e) {

            e.printStackTrace();
        }
        System.out.println("]");
    } }
```

Second :

```
public class Second extends First implements Runnable {
    String msg="";
    public Second(First f, String s) {
        msg=s;}
@Override
    public void run() {
        synchronized (this) {
            this.display(msg);
        }
        } }
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        First f= new First();
        Second s1= new Second(f, "Welcome");
        Second s2= new Second(f, "new");
        Second s3= new Second(f, "Programmer");
        Thread t1 = new Thread(s1);
        Thread t2 = new Thread(s2);
        Thread t3 = new Thread(s3);
        t1.start();

        t2.start();

        t3.start();

    } }
```

21. Write a Java program to increase the size of an array list.

Ans:

ArrayListDemo :

```
import java.util.ArrayList;

import java.util.Scanner;

public class ArrayListDemo {

    public void demoArrayList() {

        int no = 0;

        System.out.print("Enter the size of an ArrayList : ");

        int size = new Scanner(System.in).nextInt();

        ArrayList<Integer> al = new ArrayList(size);

        System.out.println("Add " + size + " elements in ArrayList : ");

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

            no = new Scanner(System.in).nextInt();

            al.add(no);

        }

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

        System.out.println("ArrayList Elements are : ");

        System.out.println(al);

        System.out.println("-----------------------------------");
        System.out.print("Do you want to increase Size of ArrayList (Y/N) :");
        String s = new Scanner(System.in).nextLine();
        if (s.equalsIgnoreCase("Y")) {
            System.out.print("Enter the Incresed size of an ArrayList : ");
            int increSize = new Scanner(System.in).nextInt();
            al.ensureCapacity(increSize);
            System.out.println("----------------------------------");
            System.out.println("ArrayList size incresed by capacity of " + increSize + "
elements!!!!");
            System.out.println("----------------------------------");
            System.out.println("Add more " + increSize + " elements in ArrayList : ");
            for (int i = 0; i < increSize; i++) {
                no = new Scanner(System.in).nextInt();
                al.add(no);
            }
            System.out.println("----------------------------------");
            System.out.println("ArrayList Elements are : ");
            System.out.println(al);
            System.out.println("----------------------------------");
} else {
            System.out.println("----------------------------------");
            System.out.println("ArrayList Updated Elements are : ");
            System.out.println(al);
            System.out.println("----------------------------------");
        }
    }
}
```

MainClass :

```
import java.util.ArrayList;
import java.util.Scanner;
public class MainClass {
    public static void main(String[] args) {
        ArrayListDemo ald = new ArrayListDemo();
        ald.demoArrayList();
    } }
```

22. Write a Java program to replace the second element of a ArrayList with the specified element.

Ans:

ArrayListDemo :

```
import java.util.ArrayList;
import java.util.Scanner;
public class ArrayListDemo {
    ArrayList<Integer> al = new ArrayList();
    public void getElements() {
        System.out.print("Enter how many elements you want to Store : ");
        int size = new Scanner(System.in).nextInt();
        for (int i = 0; i < size; i++) {
            int no = new Scanner(System.in).nextInt();
            al.add(no);

        }

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

        System.out.println("ArrayList Elements are : ");

        System.out.println(al);

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

    }

    public void replaceElement() {

        System.out.print("Enter the new elements you want to Store : ");

        int size = new Scanner(System.in).nextInt();

        System.out.print("Enter the Position of elements where you want to Store : ");

        int size1 = new Scanner(System.in).nextInt();

        al.set(size1, size);

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

        System.out.println("ArrayList Elements are : ");

        System.out.println(al);

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

    }}
```

MainClass :

```
import java.util.ArrayList;

import java.util.Scanner;

public class MainClass {

    public static void main(String[] args) {

        ArrayListDemo ald = new ArrayListDemo();
        ald.getElements();

        ald.replaceElement();

    }}
```

23. Write a Java program to print all the elements of a ArrayList using the position of the elements.

Ans:

MainClass :

```
public class MainClass {

    public static void main(String[] args) {

        PrintArrayList p= new PrintArrayList();

        p.getElements();

        p.printElements();

    }

}
```

PrintArrayList :

```
import java.util.ArrayList;

import java.util.Scanner;

public class PrintArrayList {

    ArrayList<Integer> al = new ArrayList();

    public void getElements() {

    System.out.print("Enter how many elements you want to Store : ");

        int size = new Scanner(System.in).nextInt();

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

            int no = new Scanner(System.in).nextInt();
            al.add(no);
        } }
    public void printElements() {
        System.out.println("----------------------------------");
        System.out.println("ArrayList Elements are : ");
        System.out.println(al);
        System.out.println("----------------------------------");
    } }
```

24. Write a Java program to compare two sets and retain elements which are same on both sets.

Ans:

SetDemo :

```
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;
public class SetDemo {
    Set<String> s1 = new HashSet();
    Set<String> s2 = new HashSet();
public void getFirstSet() {
        System.out.print("Enter how many elements you want to Store in First Set : ");
        int size = new Scanner(System.in).nextInt();
        for (int i = 0; i < size; i++) {
            String no = new Scanner(System.in).nextLine();
            s1.add(no);
        } }
    public void getSecondSet() {
        System.out.print("Enter how many elements you want to Store in Second Set : ");
        int size = new Scanner(System.in).nextInt();
        for (int i = 0; i < size; i++) {
            String no = new Scanner(System.in).nextLine();
            s2.add(no);
        } }
    public void getCommonElements() {
        System.out.println("----------------------------------");
        System.out.println("Elements in First Set are : ");
        System.out.println(s1);
        System.out.println("----------------------------------");
        System.out.println("Elements in Second Set are : ");
        System.out.println(s2);
        System.out.println("----------------------------------");
        System.out.println("***********************************");
        System.out.println("Common elements in both Sets are  : ");
        s1.retainAll(s2);
        System.out.println(s1);
        System.out.println("***********************************");
    }}
```

MainClass :

```
public class MainClass {
    public static void main(String[] args) {
        SetDemo sd = new SetDemo();
        sd.getFirstSet();
        sd.getSecondSet();
        sd.getCommonElements();
    }}
```

25. Write a Java program to get a collection view of the values contained in this map.

Ans:

CollectionView :

```
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
public class CollectionView {
    Map<Integer, String> m = new HashMap();
public void getElements() {
        System.out.print("Enter how many elements you want to Store : ");
        int size = new Scanner(System.in).nextInt();
        for (int i = 0; i < size; i++) {
            String s = new Scanner(System.in).nextLine();
            m.put(i, s);
        } }
public void displayMap() {
        System.out.println("---------------------------------");
        System.out.println("Collection View is : " + m.values());

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

    }  }
```

MainClass :

```
public class MainClass {

    public static void main(String[] args) {

        CollectionView cv = new CollectionView();

        cv.getElements();

        cv.displayMap();

} }
```