Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Academic Year: 2022-2023**

---

**Name – Aksh Nishar** **SAP ID - 60004220123**

## Experiment No - 1

**AIM: To implement Java Program Structures & Simple Programs**

1.i. WAP to display hello message on screen.

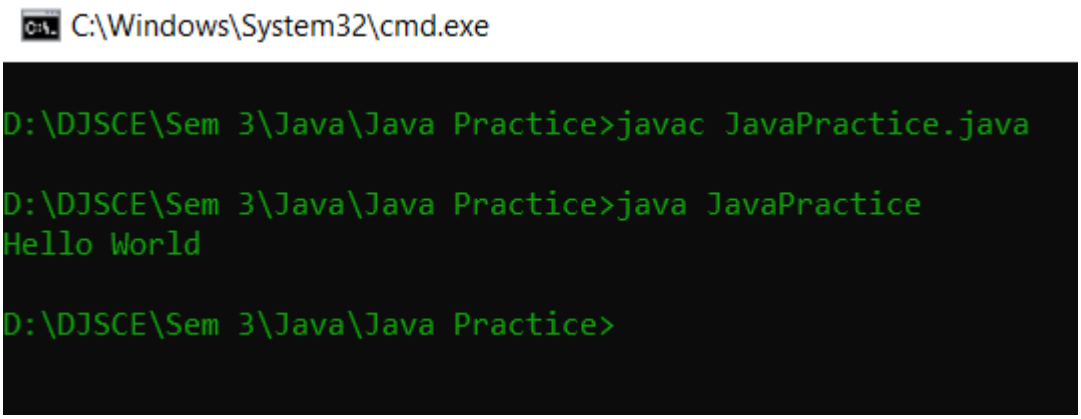
**THEORY:** In Java, System.out.println() is used to print a statement which has been passed in its argument. There are 2 printing statements in Java, the first being System.out.print() which prints the argument passed through it on the same line and the second being System.out.println() which is similar to System.out.print() method except that it moves the cursor to the next line after printing the result.

**CODE:**

JavaPractice.java > Student

```
1   /*
2    * SapId - 60004220123
3    * Name - Aksh Nishar
4    */
5
6   public class JavaPractice {
        Run | Debug
7       public static void main(String[] args) {
8           System.out.println(x: "Hello World");
9       }
10  }
```

**OUTPUT:**




**CONCLUSION:** Thus, we have written a Java Program to display Hello World on screen.

## AIM: To implement Java Program Structures & Simple Programs

1.ii. Write a Java Program that reads a positive integer from command line and count the number of digits the number (less than ten billion) has.

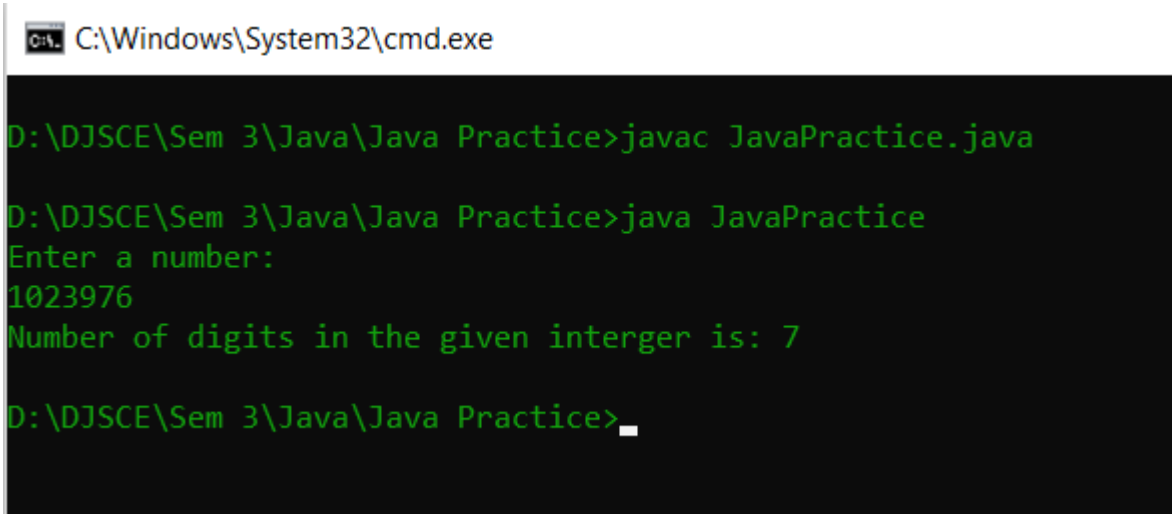
**THEORY:** The while loop in Java is a control flow statement that allows code to be executed repeatedly based on a given boolean condition. The loop goes on until the boolean condition turns false. When the number of iterations in not known to the user, they can use the while loop.

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int count = 0;
        System.out.println(x: "Enter a number: ");
        long num = scanner.nextLong();
        while (num != 0) {
            num = num / 10;
            count++;
        }

        System.out.println("Number of digits in the given interger is: " + count);

        scanner.close();
    }
}
```

## OUTPUT:




**CONCLUSION:** Thus, we have written a Java program that reads a positive integer from command line and count the number of digits the number has.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Academic Year: 2022-2023**

**Name – Aksh Nishar** **SAP ID - 60004220123**

## Experiment No - 2

**AIM:** To implement Java control statements and loops.

2.i. WAP to find roots of a Quadratic equation. Take care of imaginary values.

**THEORY:** Java has the following conditional statements:

- if to specify a block of code to be executed, if a specified condition is true
- else to specify a block of code to be executed, if the same condition is false
- else if to specify a new condition to test, if the first condition is false

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String[] z) {
        Scanner s = new Scanner(System.in);

        double a, b, c;
        double root1, root2;

        System.out.println(x: "Enter the values of coefficients a , b and c : ");
        a = s.nextDouble();
        b = s.nextDouble();
        c = s.nextDouble();
```

**Academic Year: 2022-2023**

```
19
20          double delta = b * b - 4 * a * c;
21
22          if (delta > 0) {
23              root1 = (-b + Math.sqrt(delta)) / (2 * a);
24              root2 = (-b - Math.sqrt(delta)) / (2 * a);
25              System.out.println("root1= " + root1);
26              System.out.println("root2 = " + root2);
27          } else if (delta == 0) {
28              root1 = root2 = -b / (2 * a);
29              System.out.println("root1 = root2 = " + root1);
30          } else {
31              double real = -b / (2 * a);
32              double imaginary = Math.sqrt(-delta) / (2 * a);
33              System.out.println("root1 = " + real + " + i" + imaginary);
34              System.out.println("root2 = " + real + " - i" + imaginary);
35          }
36
37          s.close();
38      }
39  }
```

**OUTPUT:**

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter the values of coefficients a , b and c :
20 10 10
root1 = -0.25 + i0.6614378277661477
root2 = -0.25 - i0.6614378277661477

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter the values of coefficients a , b and c :
10 20 10
root1 = root2 = -1.0

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter the values of coefficients a , b and c :
10 10 20
root1 = -0.5 + i1.3228756555322954
root2 = -0.5 - i1.3228756555322954

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have written a Java program to find roots of Quadratic equation.

**AIM: To implement Java control statements and loops**

2.ii. Write a menu driven program using switch case to perform mathematical operations.

**THEORY: J**ava has another conditional statement called switch. We use switch to specify many alternative blocks of code to be executed. The value for a case must be a constant or a literal. Variables are not allowed. The break statement is used inside the switch to terminate a statement sequence.

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java

/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.Scanner;

public class JavaPractice {
    Run | Debug
    public static void main(String z[]) {
        char choice;
        int x, y;
        Scanner s = new Scanner(System.in);

        System.out.println(x: "Enter any two numbers : ");
        x = s.nextInt();
        y = s.nextInt();

        System.out.println(x: "Enter a choice : ");
        System.out.println(x: "+ for Addition");
        System.out.println(x: "- for Subtraction");
        System.out.println(x: "* for Multiplication");
        System.out.println(x: "/ for Division");
        choice = s.next().charAt(index: 0);
```

```
        switch (choice) {
            case '+':
                System.out.println("The sum of " + x + " and " + y + " is " + (x + y));
                break;
            case '-':
                System.out.println("The difference of " + x + " and " + y + " is " + (x - y));
                break;
            case '*':
                System.out.println("The product of " + x + " and " + y + " is " + (x * y));
                break;
            case '/':
                System.out.println("The quotient of " + x + " and " + y + " is" +
                        (double) ((double) x / (double) y));
                break;
            case '%':
            System.out.println("The remainder of " + x + " and " + y + " is" +
                    (double) ((double) x % (double) y));
            break;
        }

        s.close();
    }
}
```

## OUTPUT:

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter any two numbers :
20 10
Enter a choice :
+ for Addition
- for Subtraction
* for Multiplication
/ for Division
% for Modulo Division
-
The difference of 20 and 10 is 10

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter any two numbers :
20 10
Enter a choice :
+ for Addition
- for Subtraction
* for Multiplication
/ for Division
% for Modulo Division
*
The product of 20 and 10 is 200

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have written a menu driven program using switch case to perform mathematical operations.

**AIM: To implement Java control statements and loops**

2.iii. WAP to display odd numbers from given range/ prime numbers from given range.

**THEORY:** In Java, for loop is used to iterate a part of the program several times. If a user knows the number of iterations, then it is recommended to use a for loop. The for statement consumes the initialization, condition and increment/decrement in one line thereby providing a shorter, easy to debug structure of looping.

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.Scanner;

public class JavaPractice {
    Run | Debug
    public static void main(String z[]) {
        Scanner s = new Scanner(System.in);
        int start, end, i, count, n;

        System.out.println(x: "Enter the start of range : ");
        start = s.nextInt();
        System.out.println(x: "Enter the end of range : ");
        end = s.nextInt();
        System.out.println(x: "The odd numbers in the range is :");

        for (i = start; i <= end; i++) {
            if (i % 2 != 0) {
                System.out.println(i);
            }
        }
```

```
24
25          System.out.println(x: "The prime numbers in the range is :");
26          for (i = start; i <= end; i++) {
27              count = 0;
28              for (n = i; n >= 1; n--) {
29                  if (i % n == 0) {
30                      count = count + 1;
31                  }
32              }
33              if (count == 2) {
34                  System.out.println(i);
35              }
36          }
37
38          s.close();
39      }
40  }
41
```

**OUTPUT:**

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter the start of range :
10
Enter the end of range :
50
The odd numbers in the range is :
11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
The prime numbers in the range is :
11
13
17
19
23
29
31
37
41
43
47

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have displayed odd/prime numbers from given range.

## AIM: To implement Java control statements and loops

2.iii. WAP to display default value of primitive data types

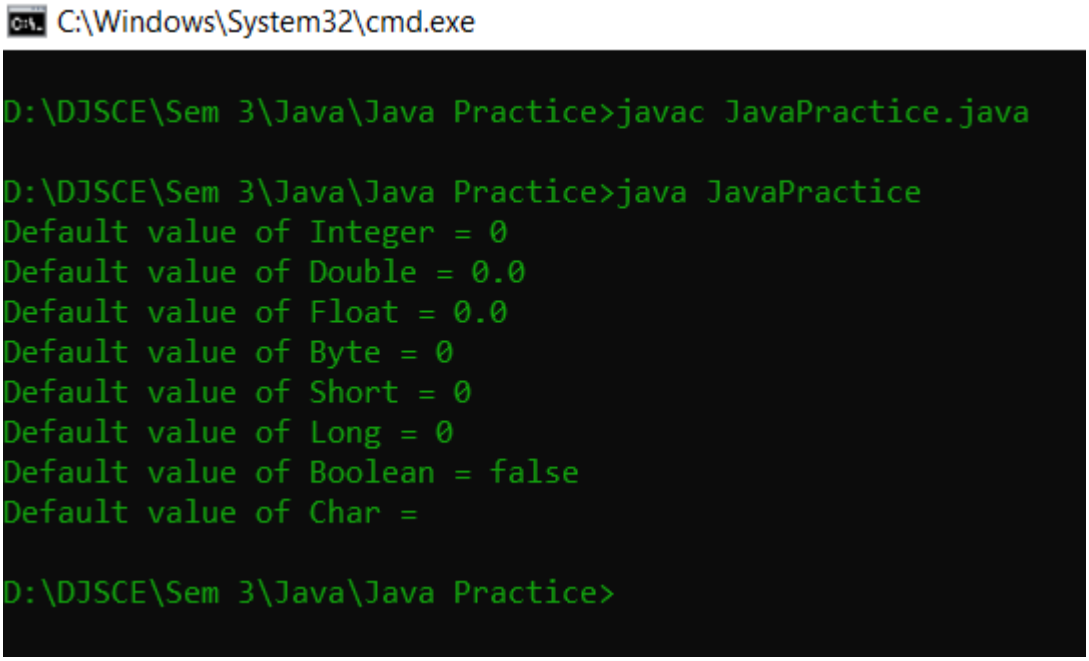
**THEORY:** Primitive data types are the building blocks of data manipulation. These are the most basic data types available in Java language. The eight primitives defined in Java are int, byte, short, long, float, double, boolean, and char – those aren't considered objects and represent raw values.

## CODE:

JavaPractice.java > JavaPractice

```java

/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    static int a;
    static double b;
    static float c;
    static byte d;
    static short e;
    static long f;
    static boolean g;
    static char h;

    Run | Debug
    public static void main(String z[]) {
        Scanner scan = new Scanner(System.in);

        System.out.println("Default value of Integer = " + a);
        System.out.println("Default value of Double = " + b);
        System.out.println("Default value of Float = " + c);
        System.out.println("Default value of Byte = " + d);
        System.out.println("Default value of Short = " + e);
        System.out.println("Default value of Long = " + f);
        System.out.println("Default value of Boolean = " + g);
        System.out.println("Default value of Char = " + h);

        scan.close();
    }
}

```

## OUTPUT:




**CONCLUSION:** Thus, we have written program to display default values of primitive data types.

## AIM: To implement Java control statements and loops

2.v. WAP to display the following patterns:

```
1
2 1
1 2 3
4 3 2 1
1 2 3 4 5
6 5 4 3 2 1
1 2 3 4 5 6 7
```

```
  A
 CB
FED
JIHG
```

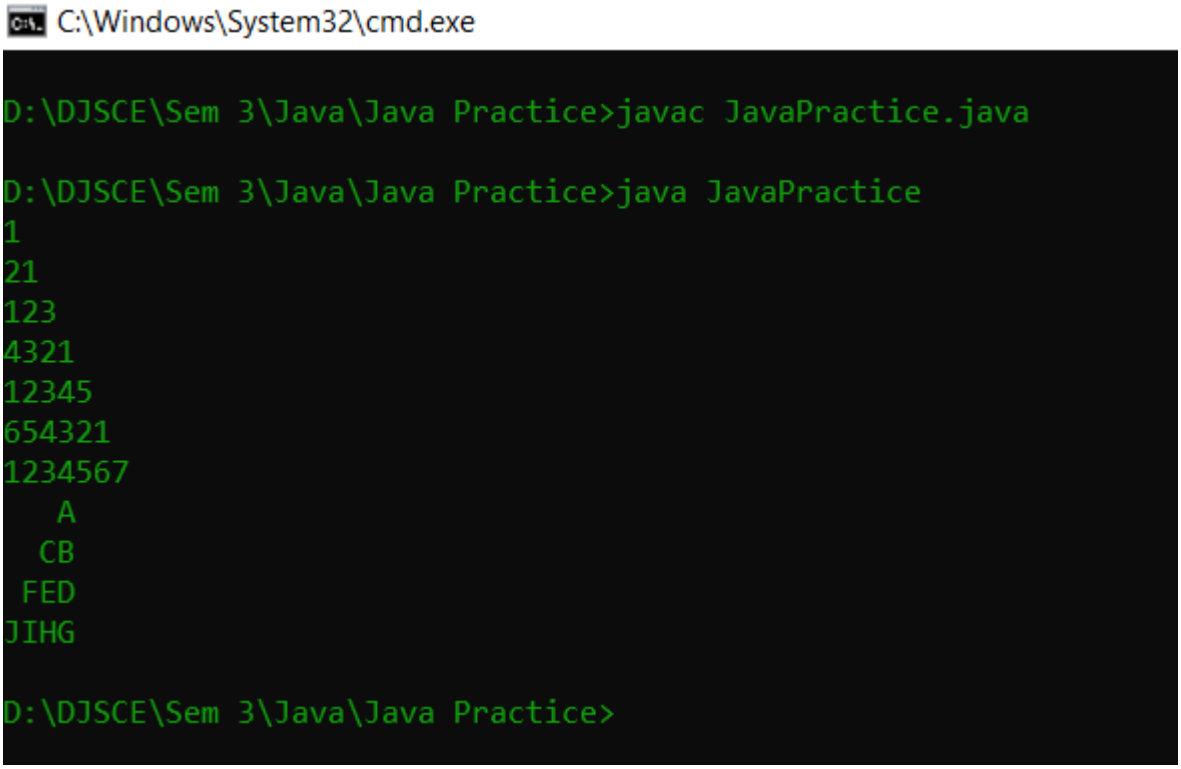
**THEORY:** If a loop exists inside the body of another loop, it's called a nested loop. That is why nested loops are also called as “loop inside loop“.

**CODE:**

JavaPractice.java > JavaPractice

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
public class JavaPractice {
    Run | Debug
    public static void main(String args[]) {
        int i, j, ch = 'A';

        for (i = 1; i <= 7; i++) {
            if (i % 2 != 0) {
                for (j = 1; j <= i; j++) {
                    System.out.print(j);
                }
            } else {
                for (j = i; j >= 1; j--) {
                    System.out.print(j);
                }
            }
            System.out.println();
        }

        for (i = 1; i <= 4; i++) {
            for (j = 4 - i; j >= 1; j--) {
                System.out.print(s: " ");
            }
            for (j = i; j > 0; j--) {
                System.out.print((char) (ch - 1 + j));
            }
            ch += i;
            System.out.println();
        }
    }
}
```

## OUTPUT:




**CONCLUSION:** Thus, we have written Java programs to display given patterns.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Academic Year: 2022-2023**

**Name – Aksh Nishar** **SAP ID - 60004220123**

## Experiment No - 3

**AIM: To implement Arrays**

3.i. WAP to find whether the entered 4 digit number is vampire or not. Combination of digits from this number forms 2 digit number. When they are multiplied by each other we get the original number. (1260=21*60, 1395=15*93, 1530=30*51)

**THEORY:** An array is a container object that holds a fixed number of values of a single type. The length of an array is established during the creation of the array.It is used to store multiple values in a single variable instead of declaring multiple variables fro each value.

**CODE:**

JavaPractice.java > ...

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "Enter a four digit number: ");

        int i, j, k, l, temp, num;
        num = sc.nextInt();
        temp = num;
        int[] arr = new int[4];

        for (i = 0; i < 4; i++) {
            arr[i] = temp % 10;
            temp /= 10;
        }

```

```
        for (i = 0; i < 4; i++) {
            for (j = 0; j < 4; j++) {
                if (i != j) {
                    for (k = 0; k < 4; k++) {
                        if (k != i && k != j) {
                            for (l = 0; l < 4; l++) {
                                if (l != i && l != j && l != k) {
                                    if ((10 * arr[i] + arr[j]) * (10 * arr[k] + arr[l]) == num) {
                                        System.out.print(num + " is a Vampire Number");
                                        System.exit(status: 0);
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        System.out.print(num + " is not a Vampire Number");

        sc.close();
    }
}

```

**OUTPUT:**

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter a four digit number: 4021
4021 is not a Vampire Number
D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter a four digit number: 1260
1260 is a Vampire Number
D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have written a program to find whether a 4 digit number is Vampire or not.

## AIM: To implement Arrays

3.ii WAP to display the following using irregular arrays

1
2 3
4 5 6

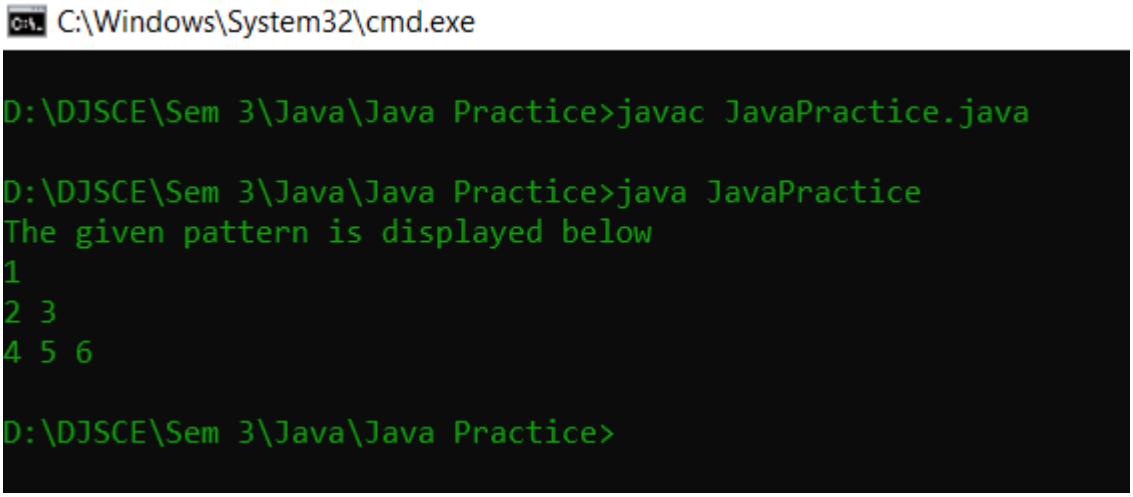
**THEORY:** Irregular arrays or jagged arrays is a group of arrays where each array can be of different sizes. We can create a 2D array with variable number of columns in each row.

**CODE:**

JavaPractice.java > JavaPractice

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
public class JavaPractice {
    Run | Debug
    public static void main(String[] args) {
        int r = 3;
        int arr[][] = new int[r][];

        for (int i = 0; i < arr.length; i++)
            arr[i] = new int[i + 1];
        int count = 1;

        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++)
                arr[i][j] = count++;
        }

        System.out.println(x: "The given pattern is displayed below ");
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr[i].length; j++)
                System.out.print(arr[i][j] + " ");
            System.out.println();
        }
    }
}
```

## OUTPUT:




**CONCLUSION:** Thus, we have displayed the given pattern using irregular arrays.

## AIM: To implement Arrays

3.iii. Write a program that queries a user for the no.: of rows and columns representing students and their marks.
Reads data row by row and displays the data in tabular form along with the row totals, column totals and grand total
Hint : For the data 1, 3, 6, 7, 9, 8 the output is

| | | | |
|---|---|---|---|
| 1 | 3 | 6 | 10 |
| 7 | 9 | 8 | 24 |
| 8 | 12 | 14 | 34 |

**THEORY:** Multidimensional arrays is an array of arrays. Each element of a multidimensional array is an array itself. The total number of elements that can be stored in a multidimensional array can be calculated by multiplying the size of all the dimensions.

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java

/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        int row, col, i, j, sum = 0;
        System.out.println("Enter the number of rows : ");
        row = in.nextInt();
        System.out.println("Enter the number of columns : ");
        col = in.nextInt();
        int[][] arr = new int[(row + 1)][(col + 1)];

        for (i = 0; i < row + 1; i++) {
            for (j = 0; j < col + 1; j++) {
                if (j != col && i != row) {
                    System.out.println("Enter element value : ");
                    arr[i][j] = in.nextInt();
                    sum = sum + arr[i][j];
                } else {
                    arr[i][j] = 0;
                }
            }
        }
```

```
29
30          for (i = 0; i < row + 1; i++) {
31              for (j = 0; j < col + 1; j++) {
32                  if (j != col && i != row) {
33                      arr[i][col] = arr[i][col] + arr[i][j];
34                      arr[row][j] = arr[row][j] + arr[i][j];
35                  } else {
36                      arr[row][col] = sum;
37                  }
38              }
39          }
40
41          for (i = 0; i < row + 1; i++) {
42              for (j = 0; j < col + 1; j++) {
43                  System.out.print(arr[i][j] + "\t");
44              }
45              System.out.println(x: " ");
46          }
47          in.close();
48      }
49  }
50
```

**OUTPUT:**

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter the number of rows :
3
Enter the number of columns :
3
Enter element value :
10
Enter element value :
20
Enter element value :
30
Enter element value :
40
Enter element value :
50
Enter element value :
60
Enter element value :
70
Enter element value :
80
Enter element value :
10
10      20      30      60
40      50      60      150
70      80      10      160
120     150     100     370

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have written a program that queries a user for the no. of rows and columns representing students and their marks and reads data row by row and displays the data in tabular form along with the row totals, columns total and grand total.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Academic Year: 2022-2023**

**Name – Aksh Nishar** **SAP ID - 60004220123**

## Experiment No - 4

**AIM: To implement Vectors**

4.i WAP that accepts a shopping list of items and performs the following operations: Add an item at a specified location, delete an item in the list, and print the *contents of the vector*.

**THEORY:** The Vector class implements a growable array of objects i.e. it can grow and shrink as per the requirements of the user. Unlike array, we can store n number of elements in it as there is no size limit.

**CODE:**

JavaPractice.java > JavaPractice > main(String[])

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String z[]) {
        Scanner in = new Scanner(System.in);
        System.out.print(s: "How many items are there in List : ");
        int n = in.nextInt();
        Vector v = new Vector(n, capacityIncrement: 3);

        for (int i = 0; i < n; i++) {
            System.out.printf(format: "Enter item %d : ", (i + 1));
            String str = in.next();
            v.addElement(str);
        }

        System.out.println(x: "Choose an action to perform");
        System.out.println(x: "1. Add an item ");
        System.out.println(x: "2. Delete an item ");
        System.out.println(x: "3. Display all items ");
        System.out.println(x: "4. Exit");

```

```
26
27          int choice = 1;
28          while (choice != 3) {
29              System.out.println(x: "Enter your choice : ");
30              choice = in.nextInt();
31              if (choice == 1) {
32                  System.out.println(x: "Enter the location of the item to be added : ");
33                  int pos = in.nextInt();
34                  System.out.println(x: "Enter the item : ");
35                  String item = in.next();
36                  v.add((pos - 1), item);
37                  System.out.println("Elements in the list are " + v);
38              } else if (choice == 2) {
39                  System.out.println(x: "Enter the item to be deleted : ");
40                  String items = in.next();
41                  v.remove(items);
42              } else if (choice == 3) {
43                  System.out.println("Items in the list are " + v);
44              } else {
45                  System.out.println(x: "Wrong choice");
46              }
47          }
48      }
49  }
50
```

## OUTPUT:

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java
Note: JavaPractice.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
How many items are there in List : 5
Enter item 1 : 10
Enter item 2 : 20
Enter item 3 : 30
Enter item 4 : 40
Enter item 5 : 50
Choose an action to perform
1. Add an item
2. Delete an item
3. Display all items
4. Exit
Enter your choice :
2
Enter the item to be deleted :
20
Enter your choice :
3
Items in the list are [10, 30, 40, 50]
```

**CONCLUSION:** Thus, we have written program on vectors to add, delete and display items.

## AIM: To implement Vectors

4.ii Write a java programs to find frequency of an element in the given Vector array.

**THEORY:** Vectors in java supports constructors and methods. Some of the constructors are
• vector()-It constructs an empty vector with the default size as 10.
• vector(int initialCapacity)- It constructs an empty vector with the specified initial capacity and with its capacity increment equal to zero.
• vector(int initialCapacity, int capacityIncrement)- It constructs an empty vector with the specified initial capacity and capacity increment.
• Vector( Collection c)- It constructs a vector that contains the elements of a collection c. Some of the methods of vector class are:
• add()-It is used to append the specified element in the given vector.
• size()-It is used to get the number of components in the given vector.
• remove()-It is used to remove the specified element from the vector. If the vector does not contain the element, it is unchanged.
• get()-It is used to get an element at the specified position in the vector.

## CODE:

JavaPractice.java > ...

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String[] args) {
        int no, i;
        Scanner sc = new Scanner(System.in);
        System.out.print(s: "No of Elements in Vector : ");
        int n = sc.nextInt();
        Vector v = new Vector(n);

        for (i = 0; i < n; i++) {
            no = i + 1;
            System.out.print("element " + no + " : ");
            String element = sc.next();
            v.addElement(element);
        }

        System.out.println("Vector : " + v);
        System.out.print(s: "Find frequency of element : ");
        String z = sc.next();
        int count = Collections.frequency(v, z);
        System.out.println("Find frequency of element " + z + " is " + count);
    }
}
```

## OUTPUT:

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java
Note: JavaPractice.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
No of Elements in Vector : 10
element 1 : 20
element 2 : 10
element 3 : 20
element 4 : 30
element 5 : 10
element 6 : 20
element 7 : 20
element 8 : 30
element 9 : 40
element 10 : 60
Vector : [20, 10, 20, 30, 10, 20, 20, 30, 40, 60]
Find frequency of element : 10
Find frequency of element 10 is 2

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have program to find the frequency of number in Vector array.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Academic Year: 2022-2023**

**Name – Aksh Nishar** **SAP ID - 60004220123**

## Experiment No - 5

**AIM: To implement Strings**

5.i. WAP to check if 2 strings are Meta strings or not. Meta strings are the strings which can be made equal by exactly one swap in any of the strings. Equal string are not considered here as Meta strings.

Example: str1 = "geeks", str2 = "keegs"
By just swapping 'k' and 'g' in any of string, both will become same.
Example: str1 = "Converse", str2 = "Conserve"
By just swapping 'v' and's' in any of string, both will become same.

**THEORY:** string is basically an object that represents sequence of char values. An array of characters works same as Java string.
Example: char[] ch={'H','a','r','s','h'};
String s=new String(ch);
Is the same as String s="Harsh";

**CODE:**

**Academic Year: 2022-2023**

JavaPractice.java > JavaPractice

```
1
2   /*
3    * SapId - 60004220123
4    * Name - Aksh Nishar
5    */
6   import java.util.*;
7
8   public class JavaPractice {
9       static String swap(String str, int i, int j) {
10          char arr[] = str.toCharArray();
11          char temp = arr[i];
12          arr[i] = arr[j];
13          arr[j] = temp;
14          return String.valueOf(arr);
15      }
16
```

```
Run | Debug
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    String str1, str2, temp;
    int i, j, n = 0;
    System.out.print(s: "Enter first word: ");
    str1 = in.nextLine();
    System.out.print(s: "Enter second word: ");
    str2 = in.nextLine();
    if (str1.length() == str2.length()) {
        for (i = 0; i < str1.length(); i++) {
            for (j = 0; j < str1.length(); j++) {
                temp = swap(str1, i, j);
                if (temp.equals(str2)) {
                    System.out.println(x: "They are a pair of Meta strings.");
                    n = 1;
                    break;
                }
            }
            if (n == 1) {
                break;
            }
        }
    }
    if (n == 0) {
        System.out.println(x: "They are not a pair of Meta strings.");
    }
}
}
```

## OUTPUT:

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter first word: Geeks
Enter second word: For
They are not a pair of Meta strings.

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice
Enter first word: Geeks
Enter second word: Geeks
They are a pair of Meta strings.

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have written a program to check if string is Meta string or not.

## AIM: To implement Strings

5.ii. Write a java program to count number of alphabets, digits, special symbols, blank spaces and words from the given sentence. Also count number of vowels and consonants.

**THEORY:** Java string class has many useful methods to perform operations on sequence of char values. Some of the are listed below:

• int length()-It returns string length

• boolean contains(CharSequence s)- It returns true or false after matching the sequence of char value.

• String toUpperCase()-Converts all of the characters in this String to upper case using the rules of the default locale. • String toLowerCase()-Converts all of the characters in this String to lowercase using the rules of the default locale.

**CODE:**

**Academic Year: 2022-2023**

JavaPractice.java > ...

```java
/*
 * SapId - 60004220123
 * Name - Aksh Nishar
 */
import java.util.*;

public class JavaPractice {
    Run | Debug
    public static void main(String[] args) {
        String line;
        Scanner in = new Scanner(System.in);
        System.out.print(s: "\nEnter the string : ");
        line = in.nextLine();
        int vowels = 0, consonants = 0, digits = 0, spaces = 0, symb = 0;
        line = line.toLowerCase();

        for (int i = 0; i < line.length(); ++i) {
            char ch = line.charAt(i);
            if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                ++vowels;
            } else if ((ch >= 'a' && ch <= 'z')) {
                ++consonants;
            } else if (ch >= '0' && ch <= '9') {
                ++digits;
            } else if (ch == ' ') {
                ++spaces;
            } else
                ++symb;
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
        System.out.println("Digits: " + digits);
        System.out.println("Blank spaces: " + spaces);
        System.out.println("Special Symbols : " + symb);
    }
}
```

**OUTPUT:**

C:\Windows\System32\cmd.exe

```
D:\DJSCE\Sem 3\Java\Java Practice>javac JavaPractice.java

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice

Enter the string : Geeks
Vowels: 2
Consonants: 3
Digits: 0
Blank spaces: 0
Special Symbols : 0

D:\DJSCE\Sem 3\Java\Java Practice>java JavaPractice

Enter the string : Hello World!!
Vowels: 3
Consonants: 7
Digits: 0
Blank spaces: 1
Special Symbols : 2

D:\DJSCE\Sem 3\Java\Java Practice>
```

**CONCLUSION:** Thus, we have program to count alphabets(vowels + consonants), digits, blank spaces and special symbols in a string.