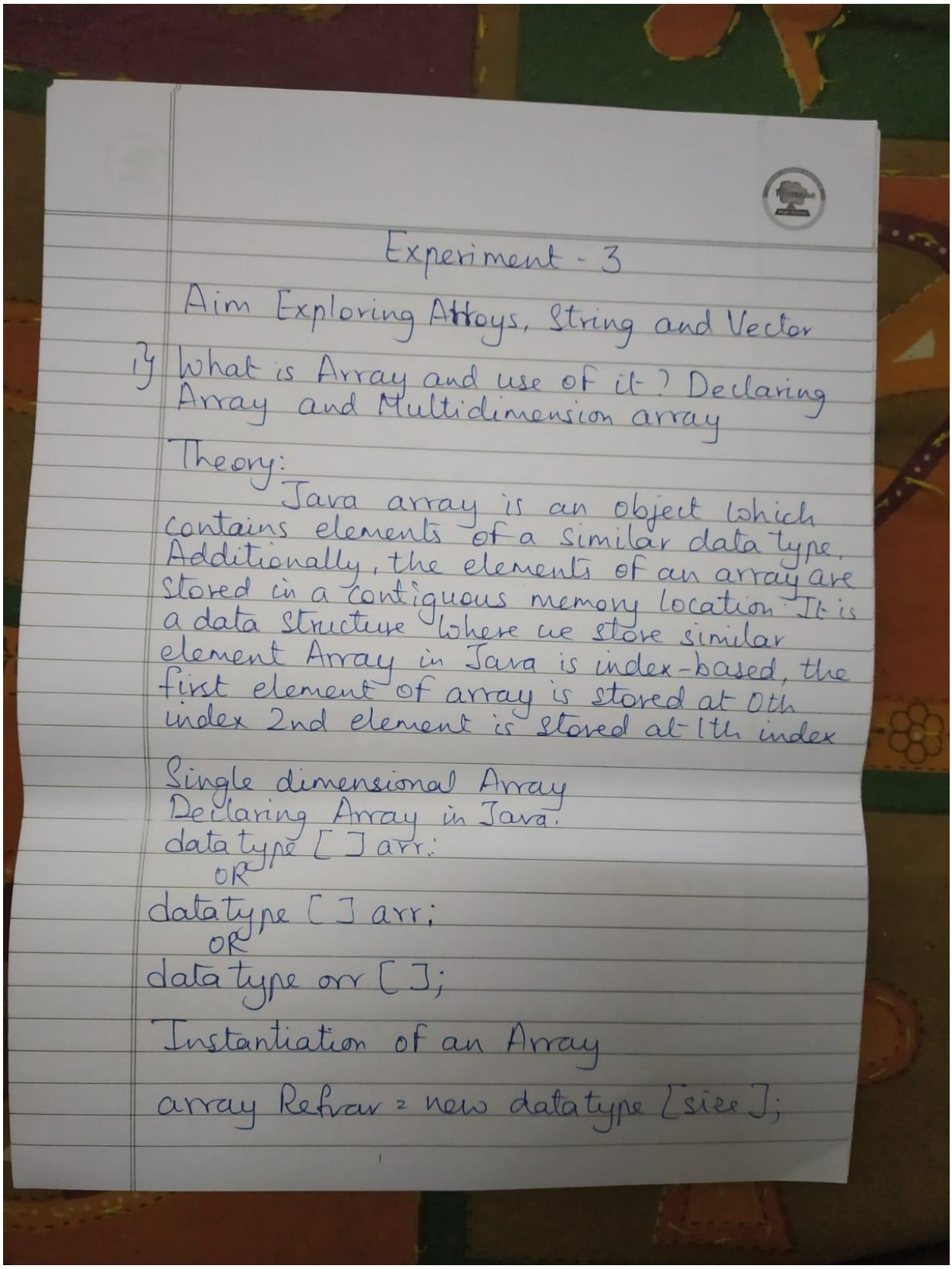
COMPUTER ENGINEERING

OOPM ODD SEM 2021-22/EXPERIMENT 3

NAME:- GAURAV AMARNANI (D7A, 67)



Experiment - 3

Aim Exploring Arrays, String and Vector

1) What is Array and use of it? Declaring Array and Multidimension array

Theory:

Java array is an object which contains elements of a similar data type. Additionally, the elements of an array are stored in a contiguous memory location. It is a data structure where we store similar element Array in Java is index-based, the first element of array is stored at 0th index 2nd element is stored at 1th index

Single dimensional Array

Declaring Array in Java.

datatype [] arr;

OR

datatype [] arr;

OR

datatype arr [];

Instantiation of an Array

array Refvar = new datatype [size];

Array is used to store data collection of same data type.

Multidimentional Array

A multidimensional Array is an array that has more than one dimension It is an array of arrays, an array that has multiple levels. The simplest multi dimensional array is 2D array. An 2D array is called Matrix

Declaring Multidimensional array

```
datatype arr [ ] [ ];
```

Initializing & accessing Array elements

```
int two d [2] [3] = {{14, 12, 10},
                     { 6, 8, 4}
                    };
```

Multi-dimensional array is used to store data for Mathematic computation, records management etc

2) What is String, list the use of string methods in Java

Java String

String is basically an object that represent sequence of char values. An array of characters works same as Java String

Syntax : String name = "value";

Java String class

Java String class provides a lot of methods to perform operation on string such as compare(), concat(), equals() etc and many more.

Methods of String in Java :-

1) length() -> Returns String length
2) Concat() -> Concatenates the string
3) replace(char old, char new) -> Replaces all the occurances of specified char value
4) toUpperCase() -> Returns a string in uppercase
5) trim() -> Removes beginning and ending space in String

6) toLowerCase() → Returns a String in lower case

7) charAt(int index) → Returns char value for particular index

Difference between String & String buffer

	String	String Buffer
1)	String class is immutable	String Buffer class is mutable
2)	String is slow	String Buffer is fast
3)	String class overrides the equal() method	String Buffer class does not override equal() Method
4)	Slower while peforming concatenation	faster with compare to String class

3) String Buffer class, Declaring string buffer and using it list and mention use of any five functions

String Buffer class - Java String Buffer class is used to create mutable (Modifiable) String objects. The String Buffer class in Java is the same as String class except it is mutable ie it can be changed

Syntax:
String Buffer variable = new StringBuffer( );

Constructor of String Buffer class:-
1) String Buffer( ) Creates an empty stringBuffer with capacity of 16
2) String Buffer (String) - creates a string with specified String
3) String Buffer (int capacity) - Creates an empty String buffer with specified capacity of length

Method of String Buffer:-
1) append (String s) → used to append specified String with this String

2) reverse( ) -> used to reverse the String
3) Capacity( ) -> used to return current
4) length( ) -> used to return length of string
5) replace (int star index, int end index, String str)
-> used to replace, string, from start index & end, index.

4) What is Vector? How to declare and use? Difference between Arraylist & Vector

→ Vector: Vector is like the dynamic array which can grow or shrink its size. Unlike array we can store n-number of elements in it as there is no size limit. It is part of Java collection framework since Java 1.2. It is found in Java Util Package

```
Vector<Type> Vector = new vector<>();
```

Method of vector class -

1) add() → used to append specified elements in given vector
2) Capacity() → Used to get current capacity of its vector
3) Clear() → used to delete all element from this vector
4) Clone() → Return a clone of this vector
5) get() → used to get an element at specified position in vector

Arraylist: Java Arraylist class used as dynamic array for storing the element. It is like an array, but there is no size limit, we can add or remove element anytime

Difference between Arraylist and Vector

Arraylist	Vector
1) Arraylist is non synchronized	Vector is Synchronized
2) Arraylist is faster the vector	Vector is slower as compare to Arraylist
3) Arraylist uses the Iterator interface to transverse element	Vector can used iterator interface or Enumeration interface to transverse.
4) Arraylist increment 50% of current array size if no. of element exceeds from capacity	Vector increment 100% mean double the array size if no of elements exceeds from capacity

Conclusion :- Thus, we implemented Arrays, String String buffer and vector with various function in a program.

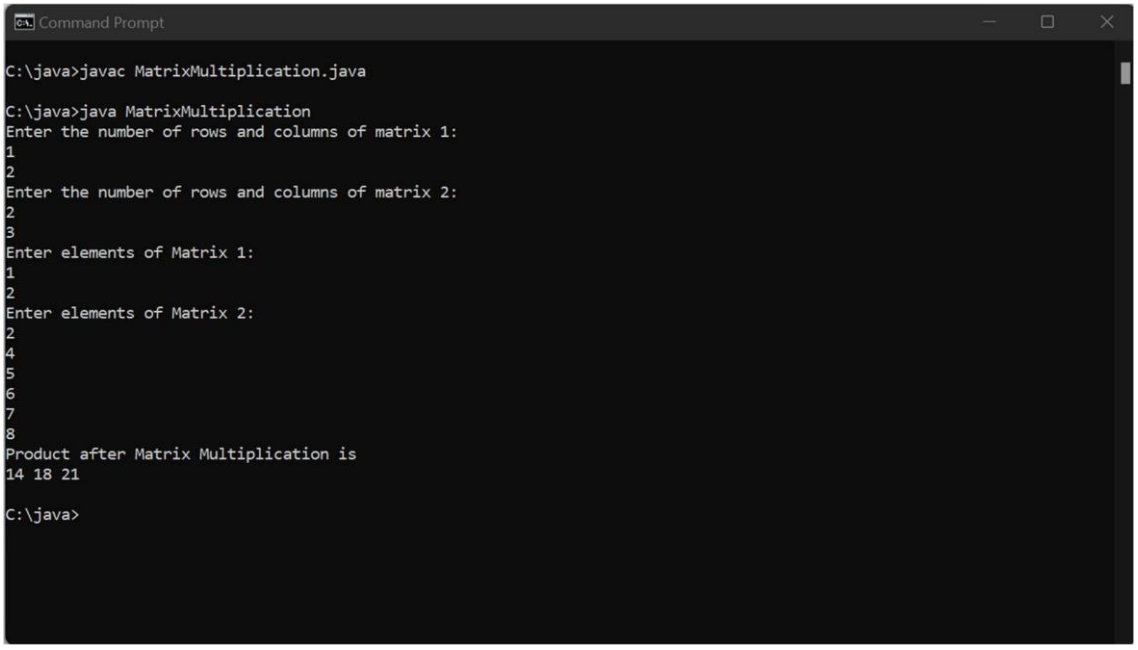
Program:

Program 1

Aim: To perform Matrix Multiplication Code-

```
import java.util.*;

class MatrixMultiplication {
public static void main( String args [ ] ) {
Scanner sc=new Scanner(System.in); int i,j,k,r,c,r1,c1,r2,c2;
System.out.println("Enter the number of rows and columns of matrix 1:"); r1=sc.nextInt();
c1=sc.nextInt();
System.out.println("Enter the number of rows and columns of matrix 2:"); r2=sc.nextInt();
c2=sc.nextInt();
if(c1==r2) {
r=r1; c=c2;
int [ ] [ ] a=new int [r1][c1];
int [ ] [ ] b=new int [r2][c2];
int [ ] [ ] product=new int [r][c];
System.out.println("Enter elements of Matrix 1:");
for(i=0;i<r1;i++) {
for(j=0;j<c1;j++) {
a[i][j]=sc.nextInt();
}
}
System.out.println("Enter elements of Matrix 2:");
for(i=0;i<r2;i++) {
for(j=0;j<c2;j++) {
b[i][j]=sc.nextInt();
}
}
for(i=0;i<r;i++) {
for(j=0;j<c;j++) {
for(k=0;k<c1;k++)
product[i][j]+=a[i][k]*b[k][j];
}
}
System.out.println("Product after Matrix Multiplication is");
for(i=0;i<r;i++) {
for(j=0;j<c;j++) {
System.out.print(product[i][j]+" ");
}
System.out.println();
}
}
else
System.out.println("Order Mismatch! Matrix Multiplication not possible!!");
}
}
```

Output:



Conclusion-

An array is initialized with all elements as zero as seen here.

Program:

Program 2

Aim: To use functions of string
Code-

```
import java.lang.*;
import java.util.*;
class Function {
public static void main(String[] args) {
String s5;
Scanner in=new Scanner(System.in);
System.out.println("The string s1 is:");
String s1=in.next();
System.out.println("The length of string is:");
System.out.println(s1.length());
System.out.println("Checking whether the character starts with z");
System.out.println(s1.startsWith("z"));
System.out.println("Checking whether the character ends with o");
System.out.println(s1.endsWith("o"));
System.out.println("Using substring(int beginIndex) we get:");
System.out.println(s1.substring(3));
System.out.println("Using substring(int beginIndex,int endIndex) we get:");
System.out.println(s1.substring(3,6));
System.out.println("Using charAt(int index):");
System.out.println(s1.charAt(2));
System.out.println("The string s2 is:");//hello
String s2=in.next();
System.out.println("Using toUpperCase() we get:");
System.out.println(s2.toUpperCase());
System.out.println("The string s3 is:");
String s3=in.next();
System.out.println("Using toLowerCase() we get");
System.out.println(s3.toLowerCase());
System.out.println("The string s4 is:");
String s4=in.next();
System.out.println("Using trim()");
System.out.println(s4.trim());
System.out.println("Using compareTo() we get:");
System.out.println(s1.compareTo(s2));
System.out.println("Using equals() we get:");
System.out.println(s1.equals(s2));
System.out.println("Using equalsIgnoreCase() we get:");
System.out.println(s2.equalsIgnoreCase(s3));
System.out.println("Using compareToIgnoreCase() we get:");
System.out.println(s2.compareToIgnoreCase(s3));
s5=s1.concat(s2);
System.out.println("Using concat() we get:");
System.out.println(s5);
}
}
```

Output:

Conclusion-

Different functions of String ,their functionalities and their syntax is learned.

Program:

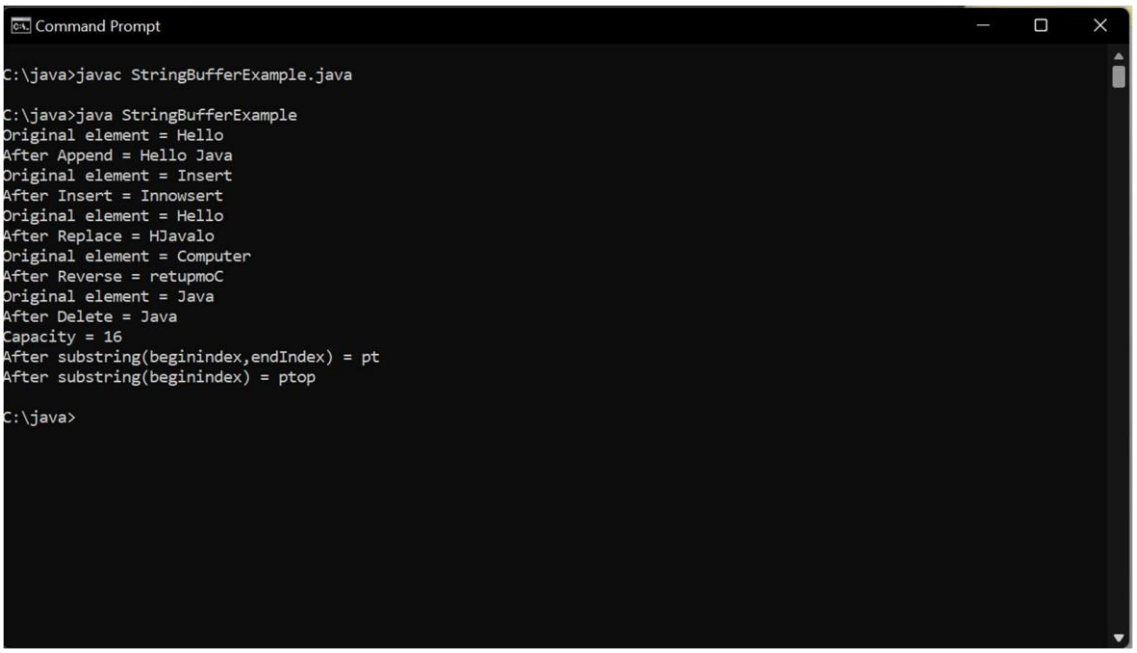
Program 3

Aim- To use functions of StringBuffer Class

Code-

```
import java.lang.*;
import java.util.*;
class StringBufferExample {
public static void main(String args[]) {
StringBuffer sb=new StringBuffer("Hello ");
System.out.println("Original element = "+sb);
sb.append("Java");
System.out.println("After Append = "+sb);

StringBuffer sb1=new StringBuffer("Insert ");
System.out.println("Original element = "+sb1);
sb1.insert(2,"now");
System.out.println("After Insert = "+sb1);

StringBuffer sb2=new StringBuffer("Hello");
System.out.println("Original element = "+sb2);
sb2.replace(1,3,"Java");
System.out.println("After Replace = "+sb2);

StringBuffer sb3=new StringBuffer("Computer");
System.out.println("Original element = "+sb3);
sb3.reverse();
System.out.println("After Reverse = "+sb3);

StringBuffer sb4=new StringBuffer("Java");
System.out.println("Original element = "+sb4);
sb.delete(1,3);
System.out.println("After Delete = "+sb4);
StringBuffer sb5=new StringBuffer();
System.out.println("Capacity = "+sb5.capacity());

StringBuffer sb6=new StringBuffer("Laptop");
System.out.println("After substring(beginindex,endIndex) = "+sb6.substring(2,4));
System.out.println("After substring(beginindex) = "+sb6.substring(2));
}
}
```

Output:



Conclusion-

Learned to use various functions of String Buffer class.

Program:

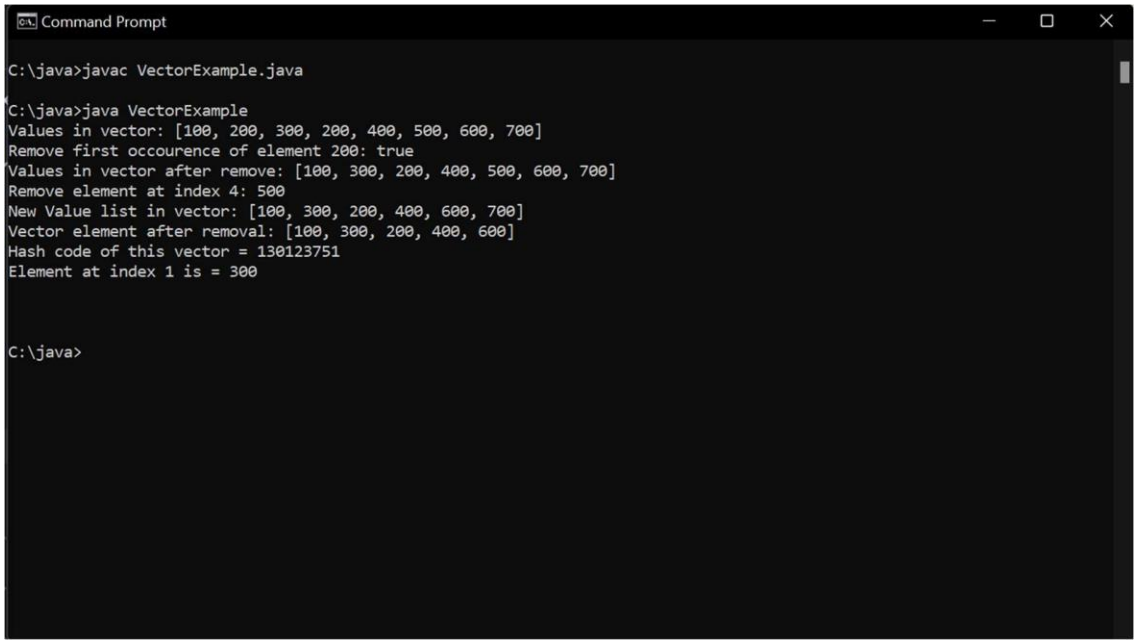
Program 4

Aim- To use functions of StringBuffer Class

Code-

```
import java.lang.*;
import java.util.*;
public class VectorExample {
      public static void main(String args[]) {
       Vector<Integer> in = new Vector<>();
       in.add(100);
       in.add(200);
       in.add(300);
       in.add(200);
       in.add(400);
       in.add(500);
       in.add(600);
       in.add(700);
       System.out.println("Values in vector: " +in);
       System.out.println("Remove first occourence of element 200: "+in.remove((Integer)200));
       System.out.println("Values in vector after remove: " +in);
       System.out.println("Remove element at index 4: " +in.remove(4));
       System.out.println("New Value list in vector: " +in);
       in.removeElementAt(5);
       System.out.println("Vector element after removal: " +in);
       System.out.println("Hash code of this vector = "+in.hashCode());
       System.out.println("Element at index 1 is = "+in.get(1));
       System.out.println("\n");
        }
}
```

Output:



Conclusion-

Learned to use various functions of Vector class.