	Hope Foundation's **Finolex Academy of Management and Technology, Ratnagiri** **Department of MCA**		
Subject name: Java Lab (SBL)			Subject Code: ITC304
Class	SEIT	Semester –III (CBCGS)	Academic year: 2022-23
Name of Student			**QUIZ Score :**
Roll No		Experiment No.	02
Title: **To study the concepts of classes, objects, arrays and Strings and implement programs based on these concepts.**			

1. Lab objectives applicable:

LOB2 - To understand the importance of Classes & objects along with constructors, Arrays, Strings and vectors

2. Lab outcomes applicable:

1. **LO1 -** Explain the fundamental concepts of Java Programing.
2. **LO2** - Use the concepts of classes, objects, members of a class and the relationships among them needed for a finding the solution to specific problem.

3. Learning Objectives:

1. To understand how to create classes and objects.
2. To work on arrays, strings and vectors.

4. Practical applications of the assignment/experiment: Concepts are used in almost every java programs.

5. Prerequisites:

1. Understanding of C programming.

6. Minimum Hardware Requirements:

1. P-IV PC with 2GB RAM, 500GB HDD, NIC

7. Software Requirements:

1. Windows / Linux operating systems, Java Developer Kit (1.8 or higher), Notepad++, Java Editors like Netbeans or Eclipse

8. Quiz Questions (if any): (Online Exam will be taken separately batch-wise, attach the certificate/ Marks obtained)

1. What are classes and objects?
2. How arrays in java and C differ?

What is the significant feature of Java Strings?

9. Experiment/Assignment Evaluation:

Sr. No.	Parameters	Marks obtained	Out of
1	Technical Understanding (Assessment may be done based on Q & A **or** any other relevant method.) Teacher should mention the other method used -		6
2	Lab Performance		2
3	Punctuality		2
Date of performance (DOP)		**Total marks obtained**	**10**

Signature of Faculty

10. Theory:

Java-Creating User Defined classes and objects

Class in Java

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- **fields**
- **methods**
- **constructors**
- **blocks**
- **nested class and interface**

Object in Java

An entity that has state and behavior is known as an object e.g. chair, bike, marker, pen, table, car etc. It can be physical or logical (tangible and intangible). The example of intangible object is banking system.

An object has three characteristics:

- **state:** represents data (value) of an object.
- **behavior:** represents the behavior (functionality) of an object such as deposit, withdraw etc.
- **identity:** Object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. But, it is used internally by the JVM to identify each object uniquely.

For Example: Pen is an object. Its name is Reynolds, color is white etc. known as its state. It is used to write, so writing is its behavior.

Object is an instance of a class. Class is a template or blueprint from which objects are created. So object is the instance (result) of a class.

Object Definitions:

- Object is *a real world entity*.
- Object is *a run time entity*.
- Object is *an entity which has state and behavior*.
- Object is *an instance of a class*.

Java-Defining and accessing/manipulating String objects

Strings, which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects.The Java platform provides the String class to create and manipulate strings.

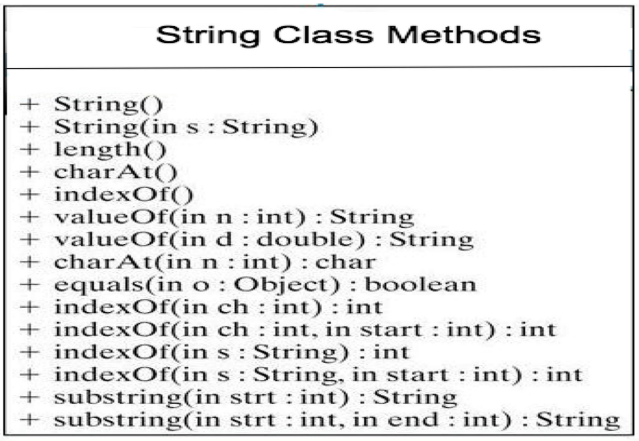
Creating Strings

The most direct way to create a string is to write −

```
String greeting = "Hello world!";
```

Whenever it encounters a string literal in your code, the compiler creates a String object with its value in this case, "Hello world!'.

As with any other object, you can create String objects by using the new keyword and a constructor. The String class has 11 constructors that allow you to provide the initial value of the string using different sources, such as an array of characters.



Java-Defining and manipulating arrays:

Java provides a data structure, the **array**, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables.

This tutorial introduces how to declare array variables, create arrays, and process arrays using indexed variables.

Declaring Array Variables

To use an array in a program, you must declare a variable to reference the array, and you must specify the type of array the variable can reference. Here is the syntax for declaring an array variable −

Syntax

```
dataType[] arrayRefVar;   // preferred way.
or
dataType arrayRefVar[];  // works but not preferred way.
```

Note − The style **dataType[] arrayRefVar** is preferred. The style **dataType arrayRefVar[]** comes from the C/C++ language and was adopted in Java to accommodate C/C++ programmers.

Example

The following code snippets are examples of this syntax −

```
double[] myList;   // preferred way.

or

double myList[];   // works but not preferred way.
```

Creating Arrays

You can create an array by using the new operator with the following syntax −

Syntax

```
arrayRefVar = new dataType[arraySize];
```

The above statement does two things −

- It creates an array using new dataType[arraySize].
- It assigns the reference of the newly created array to the variable arrayRefVar.

Declaring an array variable, creating an array, and assigning the reference of the array to the variable can be combined in one statement, as shown below −

```
dataType[] arrayRefVar = new dataType[arraySize];
```

Alternatively you can create arrays as follows −

```
dataType[] arrayRefVar = {value0, value1, ..., valuek};
```

The array elements are accessed through the **index**. Array indices are 0-based; that is, they start from 0 to **arrayRefVar.length-1**.

Java-Defining and manipulating Vector class objects:

The **java.util.Vector** class implements a growable array of objects. Similar to an Array, it contains components that can be accessed using an integer index. Following are the important points about Vector −

- The size of a Vector can grow or shrink as needed to accommodate adding and removing items.
- Each vector tries to optimize storage management by maintaining a *capacity* and a *capacityIncrement*.
- As of the Java 2 platform v1.2, this class was retrofitted to implement the List interface.
- Unlike the new collection implementations, *Vector* is synchronized.
- This class is a member of the Java Collections Framework.

Class declaration

Following is the declaration for **java.util.Vector** class −

```
public class Vector<E>
  extends AbstractList<E>
  implements List<E>, RandomAccess, Cloneable, Serializable
```

Here <E> represents an Element, which could be any class. For example, if you're building an array list of Integers then you'd initialize it as follows −

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

Class constructors

Sr.No.	Constructor & Description
1	**Vector()** This constructor is used to create an empty vector so that its internal data array has size 10 and its standard capacity increment is zero.
2	**Vector(Collection<? extends E> c)** This constructor is used to create a vector containing the elements of the specified collection, in the order they are returned by the collection's iterator.
3	**Vector(int initialCapacity)** This constructor is used to create an empty vector with the specified initial capacity and with its capacity increment equal to zero.
4	**Vector(int initialCapacity, int capacityIncrement)** This constructor is used to create an empty vector with the specified initial capacity and capacity increment.

11. Installation Steps / Performance Steps and Results –

Q1	Write a program that would print the information (name, year of joining, salary, address) of three employees by creating a class named 'Employee'. The output should be as follows: Name Year of joining Address Robert 1994 64C- WallsStreat Sam 2000 68D- WallsStreat John 1999 26B- WallsStreat

Source code:

```
import java.util.*;
class Emplyoee
{
        String name;
        int yoj;
        String addr;

 public void getInput()
{
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter Name :: ");
```

```
        name = sc.next();
        System.out.println("Enter yoj :: ");
        yoj = sc.nextInt();
        System.out.println("Enter address :: ");
        addr = sc.next();
}
public void display()
{
        System.out.print(name+"                         ");
        System.out.print(yoj+"                  ");
        System.out.print(addr+"                         ");
}
public static void main(String args[])
{
 Emplyoee e[] = new Emplyoee[3];
 for( int i=0; i<3; i++)
        {
                e[i] = new Emplyoee();
                e[i].getInput();
        }
        System.out.println("**** Data Enterd as Below ****");
        System.out.println("Name            Year of Joining                 Address");
        System.out.println();

        for( int i=0; i<3; i++)
        {

                e[i].display();
                System.out.println();
        }
}
}
```

Output:

```
        Enter Name ::
Ayush
Enter yoj ::
2003
Enter address ::
Tali_aali
Enter Name ::
```

Prasad
Enter yoj ::
2003
Enter address ::
Banda
Enter Name ::
Yash
Enter yoj ::
2003
Enter address ::
Mirya

******* Data Enterd as Below *******

Name	***Year of Joining***	***Address***
Ayush	***2003***	***Tali_aali***
Prasad	***2003***	***Banda***
Yash	***2003***	***Mirya***

Q2	Create a class 'Student' with three data members which are name, age and address. The constructor of the class assigns default values name as "unknown", age as '0' and address as "not available". It has two members with the same name 'setInfo'. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 students. Hint - Use array of objects.

Source code:

```
public class RStudent
{

  String name;
  String address;
  int age;

  RStudent()
  {
    name = "Unknown";
    age = 0;
```

```
    address = "NOt given";
  }
  public void setinfo(String n,int a)
  {
    this.name = n;
    this.age= a;
  }
  public void setinfo(String n,String d,int a)
  {
    this.name=n;
    this.address=d;
    this.age=a;
  }
  public void display()
  {
    System.out.println(name+"\t\t\t"+age+"\t\t\t"+address+"\t");
  }
  public static void main(String[] args) {
    System.out.println("Name\t\t\tAge\t\t\tAddress");

    RStudent e[] = new RStudent[10];
    e[0] = new RStudent();
    e[0].setinfo("Ayush", 19);
    e[0].display();
    e[1] = new RStudent();
    e[1].setinfo("Ayush", "Teli Aali",21);
    e[1].display();
    e[2] = new RStudent();
    e[2].setinfo("Prasad", 20);
    e[2].display();
    e[3] = new RStudent();
    e[3].setinfo("yash","MIrya", 15);
    e[3].display();
    e[4] = new RStudent();
    e[4].setinfo("SAnket","Miraj", 19);
    e[4].display();
    e[5] = new RStudent();
    e[5].setinfo("Abhi","Fansop", 10);
    e[5].display();
    e[6] = new RStudent();
    e[6].setinfo("Yash","Jaitapur", 19);
```

```
      e[6].display();
      e[7] = new RStudent();
      e[7].setinfo("Sahil", 22);
      e[7].display();
      e[8] = new RStudent();
      e[8].setinfo("Om", 16);
      e[8].display();
      e[9] = new RStudent();
      e[9].setinfo("Ved", 16);
      e[9].display();
       }
  }
```

Output:

```
Name          Age          Address

Ayush         19           NOt given
Ayush         21           Teli Aali
Prasad        20           NOt given
yash          15           MIrya
SAnket        19           Miraj
Abhi          10           Fansop
Yash          19           Jaitapur
Sahil         22           NOt given
Om            16           NOt given
Ved           16           NOt given
```

Q3	Write a java programs to add n strings in a vector array. Input new string and check whether it is present in the vector. If it is present delete it otherwise add it to the vector.

Source code:

```
        import java.util.*;
class VectorDemo
{
   public static void main(String[] args) {

      Vector<String> v = new Vector<String>();
      String s;
      char ch='y';
      Scanner sc = new Scanner(System.in);
      while(ch=='y'||ch=='y')
      {
         System.out.println("Enter a new String: ");
```

```
        s = sc.next();

        if(v.contains(s))
        {
           System.out.println(s+" is already prresent in vextor\n deleting it..");
           v.remove(s);

        }
        else
        {
           v.add(s);
           System.out.println("Vector elements are: "+v);
        }
        System.out.print("Do you want to enter string again y/n: ");
        ch = sc.next().charAt(0);
     }

     int k = v.size();
     System.out.println("Size of Vector : "+k);
     System.out.println("End of Prrgramm..");

   }
}
```

Output:

```
          Enter a new String:
Dog
Vector elements are: [Dog]
Do you want to enter string again y/n: y
Enter a new String:
Cat
Vector elements are: [Dog, Cat]
Do you want to enter string again y/n: y
Enter a new String:
Rat
Vector elements are: [Dog, Cat, Rat]
Do you want to enter string again y/n: y
Enter a new String:
Mouse
Vector elements are: [Dog, Cat, Rat, Mouse]
Do you want to enter string again y/n: y
Enter a new String:
```

Dog

Dog is already prresent in vextor

deleting it..

Do you want to enter string again y/n: n

Size of Vector : 3

End of Prrgramm..

12. Learning Outcomes Achieved

1. Students have understood concepts of classes and objects.
2. Students implemented simple java programs using Strings, arrays and objects.

13. Conclusion:

1. **Applications of the studied technique in industry**
    a. A good software design depends on proper creation of classes and objects.
2. **Engineering Relevance**
    a. Convenient handling of large data using arrays and vectors.
3. **Skills Developed**
    a. Manage data easily using classes, objects, arrays and Strings.

14. References:

[1] Stackoverflow.com. Stackoverflow.com is probably the most popular website in the programming world.
[2] Dzone.com.
[3] Java SE Technical Documentation. ...
[4] Java 2: The Complete Reference Book by Herbert Schildt
[5] The Java Language Specification Book by Bill Joy, Gilad Bracha, Guy L. Steele Jr., and James Gosling