

Data Structures and Algorithms

## Objective:

This practical is designed for you to get familiar with the usage of arrays in Java, keyboard input using java.io package and handling classes and objects as a preparation for the course *Data Structures and Algorithms in Java*.

1. Write following java program using suitable text editor in Fedora Linux OS and save it on your home directory with a suitable name. Get an idea about keyboard input using java.io package.

```
import java.io.*; // for I/O
```

```
class SimpleInput
```

```
{
    public static void main(String[] args) throws IOException
    {
        int[] arr=new int[10];    // make an array
        int nElems;              // number of items
        int j;                   // loop counter

        nElems = 10;

        String s="";
        BufferedReader input=new BufferedReader(new InputStreamReader(System.in));
        for (j=0; j<nElems; j++){
            s=input.readLine();    //read elements
            arr[j]=Integer.parseInt(s);
        }
        for (j=0; j<nElems; j++)    // display items
            System.out.print(arr[j] + " ");
    } // end main()
} // end class SimpleInput
```

2. Open a Linux terminal and compile & execute above program using following commands
  - ❖ javac YourProgramName.java
  - ❖ java YourProgramName

3. Determine the type of an array for these elements and make an array.  
2, 6, 8, 1, 3, 4, 7, 2, 2, 5, 6, 8, 9, 3, 1, 1, 2, 5, 8, 4  
Implement the problems in JAVA.

- a) Print the array.
- b) Find the maximum value of the set. Print the max value.
- c) Find the minimum value of the set. Print the minimum value.
- d) Calculate the sum of the array elements and display it.

4. Write a java program to read your name using the keyboard and print the letters of your name in reverse order.

### Working with Classes.

5. Execute the following program and examine the code carefully.

```
import java.io.*;
class Person
{
    static String name;
    static int age;
    static String address;
    public static void set(String n, int a, String add){
        name=n;
        age=a;
        address=add;
    }
    public static void display(){
        System.out.println("\nName="+name);
        System.out.println("Age="+age);
        System.out.println("Address="+address);
    }
}

public class MyDetail{
    public static void main(String arg[]) throws IOException
    {
        Person p=new Person();
        String n="";
        String add="";
        String s="";
        int a;
        BufferedReader input=new BufferedReader(new InputStreamReader(System.in));
        System.out.print("\nEnter Name=");
        n=input.readLine();
        System.out.print("\nEnter Age=");
        s=input.readLine();
        a=Integer.parseInt(s);
        System.out.print("\nEnter Address=");
        add=input.readLine();
        p.set(n,a,add);
        p.display();
    }
}
```

**Working with Constructors**

6. Following example for constructor overloading. Execute the following program and get an idea about constructor overloading.

```
class t2 {
    t2 () {
        System.out.println("empty arg called!");
    }

    t2 (int n) {
        System.out.println("int one called!");
    }

    t2 (double n) {
        System.out.println("double one called!");
    }
}

public class t1 {
    public static void main(String[] arg) {
        // creating 3 objects of t2.
        // when each object are created, Java automatically calls the right constructor
        t2 x1 = new t2();
        t2 x2 = new t2(3);
        t2 x3 = new t2(3.0);
    }
}
```

7. Following example is to identify the use of **this** key word. Try with this example.

```
class ThisDemo
{
    public ThisDemo() {
        this(10);
        System.out.println("First Constructor");
    }
    public ThisDemo(int a) {
        this(10,20);
        System.out.println("Second Constructor");
    }
    public ThisDemo( int a, int B) {
        this("Prasad");
        System.out.println("Third Constructor");
    }
    public ThisDemo(String s){
        System.out.println("Fourth Constructor");
    }
    public static void main(String args[]) {
        ThisDemo first = new ThisDemo();
    }
}
```

8. Following code is to understand the instance variables, class variables and the class method. Type this code in a text editor and save it using proper name. Then compile, execute and examines the answer.

```
class Bicycle{
    private int cadence;
    private int gear;
    private int speed;
    private int id;

    private static int numberOfBicycles;

    public Bicycle(int startCadence, int startSpeed, int startGear){
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;
        id = ++numberOfBicycles;
    }

    public int getID() {
        return id;
    }

    public static int getNumberOfBicycles() {
        return numberOfBicycles;
    }
}

public class BicycleClass{
    public static void main(String a[]){
        Bicycle b1=new Bicycle(23,40,1);
        Bicycle b2=new Bicycle(20,20,1);
        System.out.println(" Bike 1 ID="+b1.getID());
        System.out.println(" Bike 2 ID="+b2.getID());
        System.out.println("No of Bicycles="+ Bicycle.getNumberOfBicycles());
        Bicycle b3=new Bicycle(10,30,2);
        System.out.println("Bike 3 ID="+b3.getID());
        System.out.println("No of Bicycles="+ Bicycle.getNumberOfBicycles());
    }
}
```