Syeda Reeha Quasar

14114802719

4C7

Topics Covered

Arrays, Methods, Method Overloading

Lab - 3

Java Programming Lab

# **EXPERIMENT – 3.1**

## **Aim:**

Write a program to find out the array index or position where sum of numbers preceding the index is equals to sum of numbers succeeding the index.

## **Theory:**

Given, an array of size n. Find an element index that divides the array into two sub-arrays with equal sums.

Examples:

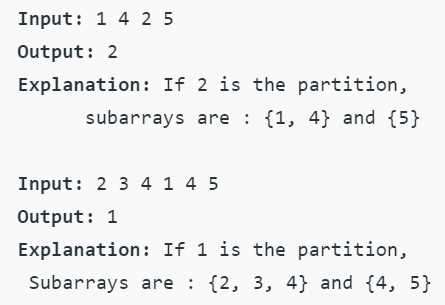
**Input:** 1 4 2 5

**Output:** 2

**Explanation:** If 2 is the partition, subarrays are : {1, 4} and {5}

**Concepts:**

* **class** keyword is used to declare a class in Java.
* **public** keyword is an access modifier that represents visibility. It means it is visible to all.
* **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method. So, it saves memory.
* **void** is the return type of the method. It means it doesn't return any value.
* **main** represents the starting point of the program.
* **String[] args** or **String args[]** is used for [command line argument](https://www.javatpoint.com/command-line-argument). We will discuss it in coming section.
* **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class. We will discuss the internal working of [System.out.println()](https://www.javatpoint.com/system-out-println-in-java) statement in the coming section.
* **Java Utils:** Resources Job Search Discussion. Java. util package contains the **collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes**. This reference will take you through simple and practical methods available in java.
* util. Java util package contains **collection framework, collection classes**, classes related to date and time, event model, internationalization, and miscellaneous utility classes. ... On importing this package, you can access all these classes and methods.
* **Scanner** is a class in **java.** **util package used for obtaining the input of** the primitive types like int, double, etc. and strings. ... To create an object of Scanner class, we usually pass the predefined object System.in, which represents the standard input stream.
* A switch statement **allows a variable to be tested for equality against a list of values**. Each value is called a case, and the variable being switched on is checked for each case.
* The input is **the data that we give to the program**. The output is the data what we receive from the program in the form of result. Stream represents flow of data or the sequence of data.



## **Source Code:**

// Program to find an element such that sum of right side element is equal to sum of left side

public class sumFragment {

// Finds an element in an array such that

// left and right side sums are equal

static int findElement(int arr[], int n) {

// Forming prefix sum array from 0

int[] prefixSum = new int[n];

prefixSum[0] = arr[0];

for (int i = 1; i < n; i++)

prefixSum[i] = prefixSum[i - 1] + arr[i];

// Forming suffix sum array from n-1

int[] suffixSum = new int[n];

suffixSum[n - 1] = arr[n - 1];

for (int i = n - 2; i >= 0; i--)

suffixSum[i] = suffixSum[i + 1] + arr[i];

// Find the point where prefix and suffix

// sums are same.

for (int i = 1; i < n - 1; i++)

if (prefixSum[i] == suffixSum[i])

return i; // idex where equal

return -1;

}

public static void main(String args[]) {

int arr[] = { 1, 4, 3, 5 };

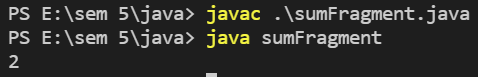
int n = arr.length;

System.out.println(findElement(arr, n));

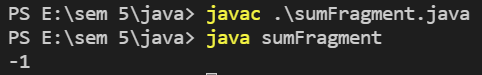
}

}

## **Output:**



int arr[] = { 1, 4, 2, 3, 5 };



# **EXPERIMENT – 3.2**

## **Aim:**

Write a program that creates and initializes a four-element int array. Calculate and display the average of its values.

## **Theory:**

Enter size of array and then enter all the elements of that array. Now using for loop we calculate sum of elements of array and hence we divide it by number of elements in array to get average.

Here is the source code of the Java Program to Calculate Sum & Average of an Array. The Java program is successfully compiled and run on a Windows system.

**Concepts:**

* **class** keyword is used to declare a class in Java.
* **public** keyword is an access modifier that represents visibility. It means it is visible to all.
* **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method. So, it saves memory.
* **void** is the return type of the method. It means it doesn't return any value.
* **main** represents the starting point of the program.
* **String[] args** or **String args[]** is used for [command line argument](https://www.javatpoint.com/command-line-argument). We will discuss it in coming section.
* **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class. We will discuss the internal working of [System.out.println()](https://www.javatpoint.com/system-out-println-in-java) statement in the coming section.
* **Scanner** is a class in **java.** **util package used for obtaining the input of** the primitive types like int, double, etc. and strings. ... To create an object of Scanner class, we usually pass the predefined object System.in, which represents the standard input stream.

## **Source Code:**

import java.util.Scanner;

public class avgArray {

public static void main(String[] args) {

double[] arr = new double[4];

double total = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter 4 elements for which you want to calculate average: ");

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

arr[i] = sc.nextDouble();

total = total + arr[i];

}

double average = total / arr.length;

/\*

\* This is used for displaying the formatted output if you give %.4f then the

\* output would have 4 digits after decimal point.

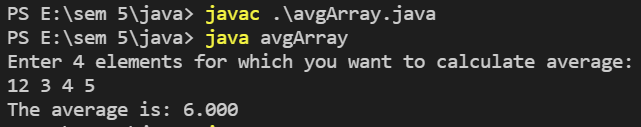
\*/

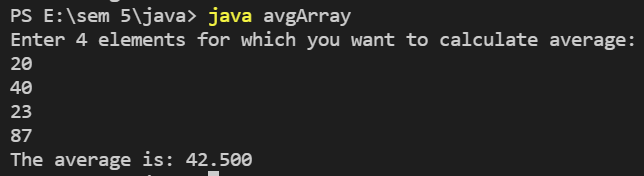
System.out.format("The average is: %.3f", average);

}

}

## **Output:**





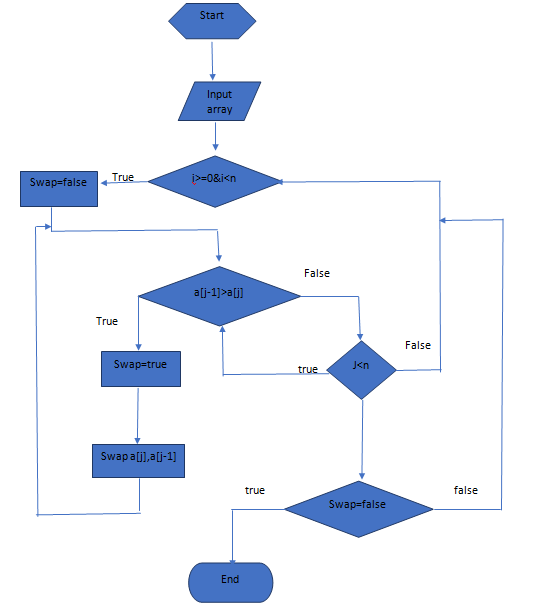
# **EXPERIMENT – 3.3**

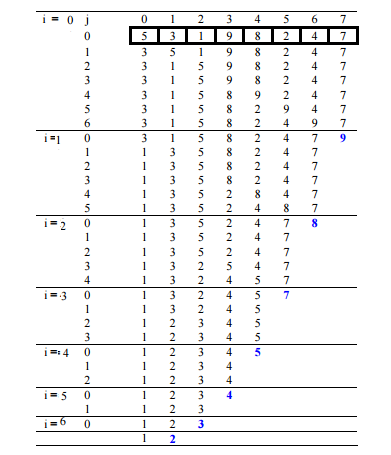
## **Aim:**

WAP using Bubble sort for sorting in ascending Order.

## **Theory:**

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in wrong order.





## **Source Code:**

import java.util.Scanner;

public class bubbleSort {

void sort(int[] arr, int n) {

int temp;

for (int i = 0; i < n - 1; i++)

for (int j = 0; j < n - i - 1; j++)

if (arr[j] > arr[j + 1]) {

temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

System.out.println("Sorted Array:");

for (int i = 0; i < n; i++)

System.out.print(arr[i] + " ");

}

public static void main(String[] args) {

bubbleSort sort = new bubbleSort();

int n;

Scanner sc = new Scanner(System.in);

System.out.print("Enter the size of array ");

n = sc.nextInt();

var arr = new int[n];

System.out.println("Enter the array");

for (int i = 0; i < n; i++)

arr[i] = sc.nextInt();

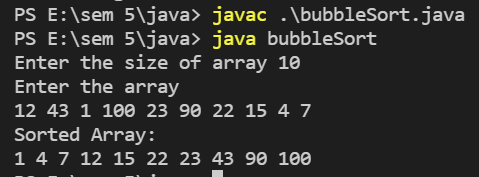
sc.close();

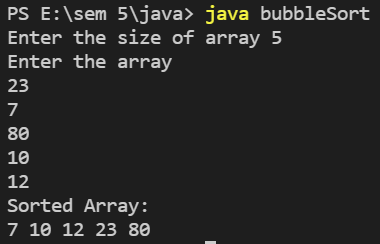
sort.sort(arr, n);

}

}

## **Output:**





# **EXPERIMENT – 3.4**

## **Aim:**

Create a java program to implement stack and queue concept.

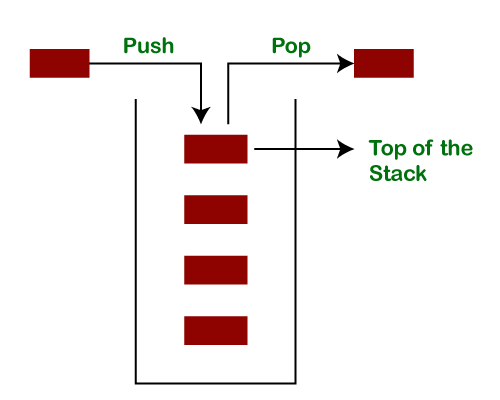
## **Theory:**

Java Stack

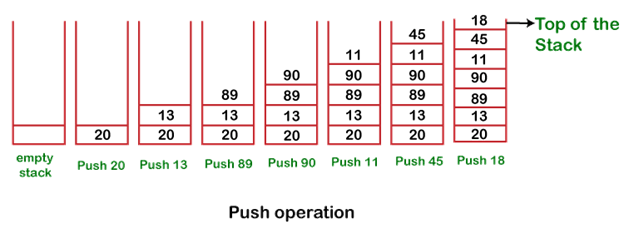
The stack is a linear data structure that is used to store the collection of objects. It is based on Last-In-First-Out (LIFO). [Java collection](https://www.javatpoint.com/collections-in-java) framework provides many interfaces and classes to store the collection of objects. One of them is the Stack class that provides different operations such as push, pop, search, etc.

In this section, we will discuss the Java Stack class, its methods, and implement the stack data structure in a [Java program](https://www.javatpoint.com/java-programs). But before moving to the Java Stack class have a quick view of how the stack works.

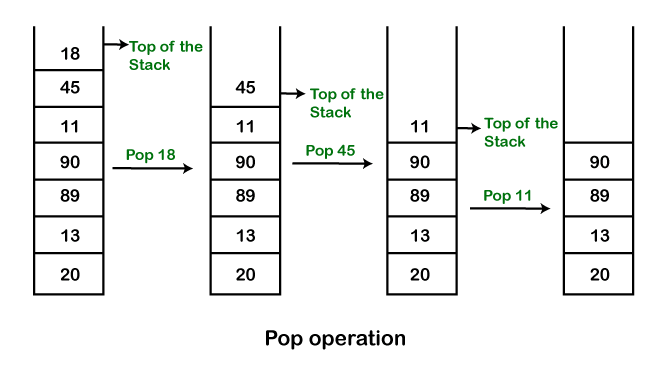
The stack data structure has the two most important operations that are push and pop. The push operation inserts an element into the stack and pop operation removes an element from the top of the stack. Let's see how they work on stack.



Let's push 20, 13, 89, 90, 11, 45, 18, respectively into the stack.



Let's remove (pop) 18, 45, and 11 from the stack.



**Queue Interface In Java**

The Queue interface present in the [java.util](https://www.geeksforgeeks.org/java-util-package-java/) package and extends the [Collection interface](https://www.geeksforgeeks.org/collections-in-java-2/) is used to hold the elements about to be processed in FIFO(First In First Out) order. It is an ordered list of objects with its use limited to insert elements at the end of the list and deleting elements from the start of the list, (i.e.), it follows the FIFO or the First-In-First-Out principle.

Being an interface the queue needs a concrete class for the declaration and the most common classes are the [PriorityQueue](https://www.geeksforgeeks.org/priority-queue-class-in-java-2/) and [LinkedList](https://www.geeksforgeeks.org/linked-list-in-java/) in Java.It is to be noted that both the implementations are not thread safe. [PriorityBlockingQueue](https://www.geeksforgeeks.org/priorityblockingqueue-class-in-java/) is one alternative implementation if thread safe implementation is needed.

## **Source Code:**

class multiply {

public static void main(String[] args) {

double num1 = Double.valueOf(args[0]);

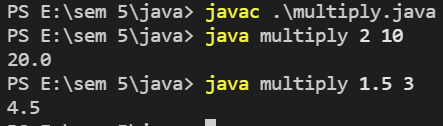
double num2 = Double.valueOf(args[1]);

System.out.println(num1 \* num2);

}

}

## **Output:**





# **EXPERIMENT – 2.5**

## **Aim:**

Write an application that accepts one command line argument, display the line of reporting if number is even or odd.

## **Theory:**

In this program, you'll learn to check if a number entered by an user is even or odd. This will be done using if...else statement and ternary operator in Java.

To understand this example, you should have the knowledge of the following [Java programming](https://www.programiz.com/java-programming) topics:

* [Java if...else Statement](https://www.programiz.com/java-programming/if-else-statement)
* [Java Scanner Class](https://www.programiz.com/java-programming/scanner)
* **Java Utils:** Resources Job Search Discussion. Java. util package contains the **collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes**. This reference will take you through simple and practical methods available in java.
* util. Java util package contains **collection framework, collection classes**, classes related to date and time, event model, internationalization, and miscellaneous utility classes. ... On importing this package, you can access all these classes and methods.
* **Scanner** is a class in **java.** **util package used for obtaining the input of** the primitive types like int, double, etc. and strings. ... To create an object of Scanner class, we usually pass the predefined object System.in, which represents the standard input stream.



## **Source Code:**

public class EvenOdd {

public static void main(String[] args) {

int num = Integer.valueOf(args[0]);

if (num % 2 == 0)

System.out.println(num + " is even");

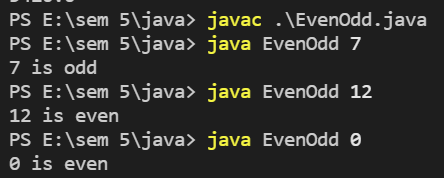
else

System.out.println(num + " is odd");

}

}

## **Output:**



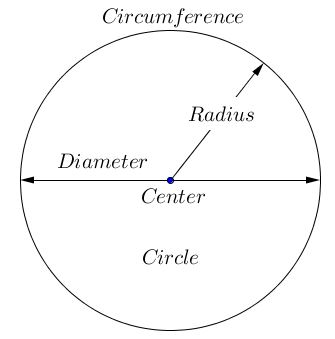


# **EXPERIMENT – 2.6**

## **Aim:**

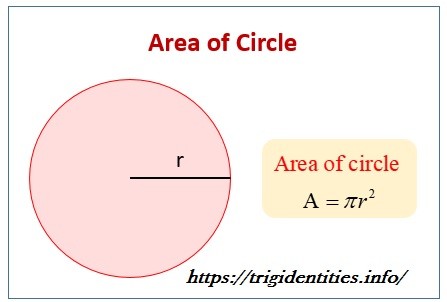
Write an application that accepts radius of a circle as its command line argument display the area.

## **Theory:**



In geometry, the area enclosed by a circle of radius r is πr2. Here the Greek letter π represents a constant, approximately equal to 3.14159, which is equal to the ratio of the circumference of any circle to its diameter.

The circumference of a circle is the linear distance around its edge.



## **Source Code:**

public class CircleArea {

public static void main(String[] args) {

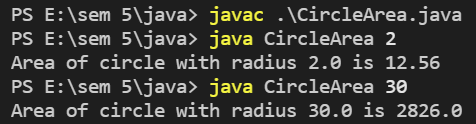
double radius = Double.valueOf(args[0]);

System.out.println("Area of circle with radius " + radius + " is " + (radius \* radius \* 3.14));

}

}

## **Output:**





# **Viva Questions**

### **1. What is a pointer and does Java support pointers?**

### Ans.

Pointer is a reference handle to a memory location. Improper handling of pointers leads to memory leaks and reliability issues hence Java doesn't support the usage of pointers.

### **2. What is the base class of all classes?**

Ans.

java.lang.Object

### **3. Does Java support multiple inheritance?**

Ans.

Java doesn't support multiple inheritance.

### **4. Is Java a pure object-oriented language?**

Ans.

Java uses primitive data types and hence is not a pure object-oriented language.

### **5. Are arrays primitive data types?**

Ans.

In Java, Arrays are objects.