**Assignment: (Core Java)**

**8 : Arrays and Strings**

**Que.1 One-Dimensional and Multidimensional Arrays**

**Ans.1**  A **one-dimensional array** is simply a list of elements of the same data type stored in contiguous memory locations.

**Syntax:**

dataType[] arrayName = new dataType[size];

example:

public class oneDArray {

public static void main(String[] args) {

int[] numbers = {10, 20, 30, 40, 50};

System.out.println("One-Dimensional Array Elements:");

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

System.out.println("Element at index " + i + ": " + numbers[i]);

}

}

}

**Que.2 String Handling in Java: String Class, StringBuffer, StringBuilder**

**Ans.2** In Java, strings are objects that represent a sequence of characters. Java provides three main classes for string handling:

* **String (Immutable)**
  + Immutable → Once created, you cannot change the value.
  + Stored in the String Constant Pool (SCP) for memory efficiency.
  + If you modify a string, a new object is created.
* **StringBuffer (Mutable, Thread-safe)**
  + Mutable → Can be modified without creating a new object.
  + Thread-safe → Multiple threads cannot access it simultaneously.
  + Slightly slower than StringBuilder.
* **StringBuilder (Mutable, Faster, Not Thread-safe)**
  + Same as StringBuffer but NOT thread-safe.
  + Faster than StringBuffer (used in single-threaded applications).

**Que.3 Array of Objects**

**Ans.3** In Java, arrays can store objects just like they store primitive data types.So, an array of objects is simply an array that holds references to class objects.

**Syntax:**

ClassName[] arrayName = new ClassName[size];

* + - This creates an array of references (not actual objects yet).
    - Each element must be initialized using new.

**Example:**

class Student {

String name;

int age;

// Constructor

Student(String name, int age) {

this.name = name;

this.age = age;

}

// Method to display student details

void display() {

System.out.println("Name: " + name + ", Age: " + age);

}

}

// Main class

public class ArrayOfObjects {

public static void main(String[] args) {

// Create an array of 3 Student objects

Student[] students = new Student[3];

// Initialize each object

students[0] = new Student("Anita", 20);

students[1] = new Student("Roshani", 21);

students[2] = new Student("Payal", 22);

// Display details of each student

System.out.println("Student Details:");

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

students[i].display();

}

}

}

**Que.4 String Methods (length, charAt, substring, etc.)**

**Ans.4** In Java, String objects provide several useful methods for manipulating and extracting information from text. Three fundamental methods are length(), charAt(), and substring().

**length() Method :**The length() method returns the number of characters in a String object. This includes letters, numbers, special characters, and spaces.

**Example:**

String text = "Hello World";

int length = text.length(); // length will be 11

**charAt() Method:** The charAt() method returns the character at a specified index within a String. In Java, string indices are zero-based, meaning the first character is at index 0, the second at index 1, and so on.

**Example:**

String word = "YAMI";

char firstChar = word.charAt(0); // firstChar will be 'Y'

char thirdChar = word.charAt(2); // thirdChar will be 'I'

**substring() Method:** The substring() method extracts a portion of a String and returns it as a new String object. It has two overloaded versions:

* substring(int beginIndex): This version returns a substring starting from beginIndex (inclusive) and extending to the end of the original string.

**Example:**

String sentence = "My name is YAMI Desai";

String sub1 = sentence.substring(15); // sub1 will be "Desai"

* substring(int beginIndex, int endIndex): This version returns a substring starting from beginIndex (inclusive) and extending up to endIndex (exclusive). The character at endIndex is not included in the resulting substring.

**Example:**

String data = "abcdefgh";  
String sub2 = data.substring(2, 5); // sub2 will be "cde" (characters at index 2, 3, and 4)