

## **MODULE – 3**

### Arrays, String and String Buffer



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# Agenda

- ✓ Introduction To Arrays
- ✓ Declaration, Initialization, Accessing an Array
- ✓ Multi –Dimensional Array
- ✓ Array of objects
- ✓ String: Creation & Operation
- ✓ String buffer: Creation & Operation



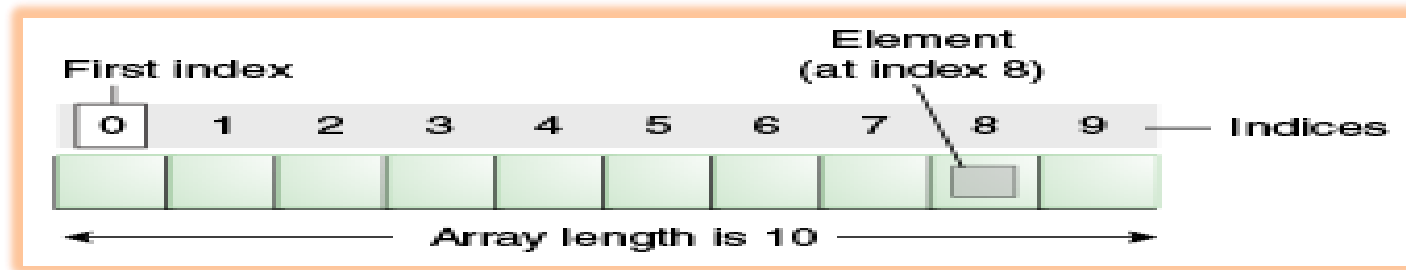
**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# Introduction to Arrays

- **Java array is an object which contains elements of a similar data type.**
- **The elements of an array are stored in a contiguous memory location.**
- **It is a data structure where we store similar elements.**
- **We can store only a fixed set of elements in a Java array.**
- **Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.**



- ✓ Each item in an array is called an element, and each element is accessed by its numerical index. As shown above, numbering begins with 0. The 9th element, for example, would therefore be accessed at index 8.

40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

<- Array Indices

Array Length = 9  
First Index = 0  
Last Index = 8



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



## Advantages:

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data efficiently.
- **Random access:** We can get any data located at an index position.

## Disadvantages:

- **Size Limit:** We can store only the fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in Java which grows automatically.



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# Declaring an Array

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[] ArrayName; // preferred way.
```

or

```
dataType ArrayName[]; // works but not preferred way
```

## Example:

The following code snippets are examples of this syntax –

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

or

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

# Creating an Array

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

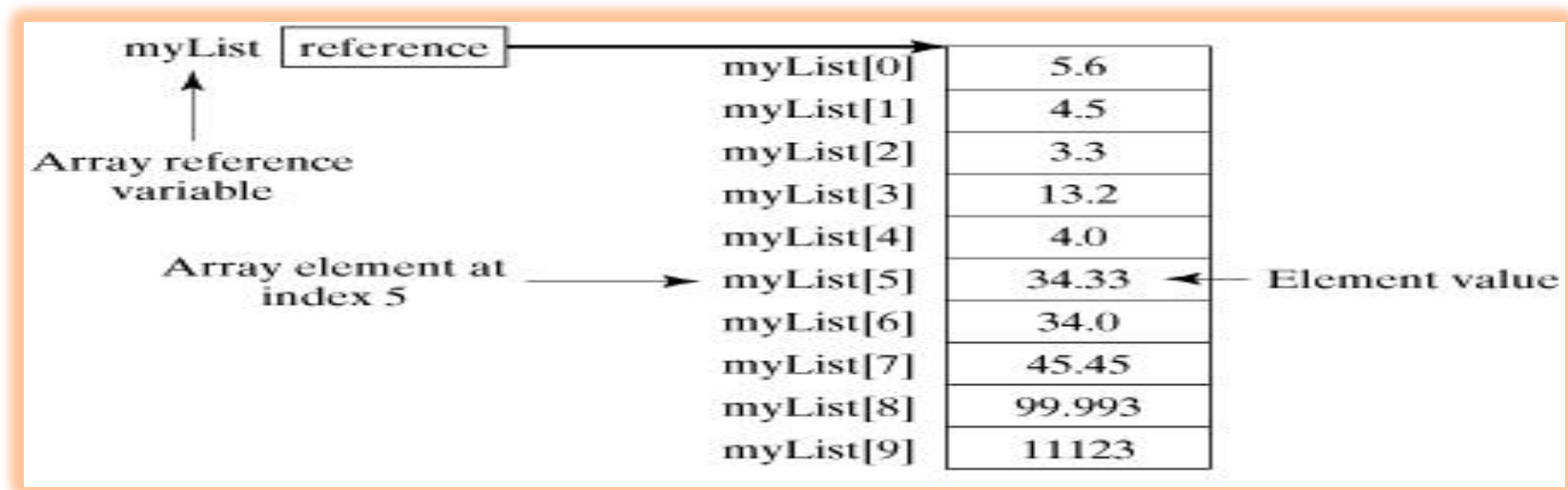
Declaring an array variable, creating an array, and assigning the reference of the array to the variable can be combined in one statement, as shown below –

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

## Example

Following statement declares an array variable, myList, creates an array of 10 elements of double type and assigns its reference to myList –

```
double[] myList = new double[10];
```



Following picture represents array myList. Here, myList holds ten double values and the indices are from 0 to 9.

Alternatively you can create arrays as follows –

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



## Example of Java Array:

Let's see the simple example of java array, where we are going to declare, instantiate, initialize and traverse an array.

```
class Testarray
{
    public static void main(String args[]){

        int a[]=new int[5]; //declaration and instantiation
        a[0]=10;             //initialization
        a[1]=20;
        a[2]=70;
        a[3]=40;
        a[4]=50;

        for(int i=0;i<5;i++) //printing array
            System.out.println(a[i]);
    }
}
```

Output:

10

20

70

40

50



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



Let's see the simple example to print this array.

//Java Program to illustrate the use of declaration, instantiation  
//and initialization of Java array in a single line

```
class Testarray1{  
    public static void main(String args[])  
    {  
  
        int a[]={33,3,4,5}; //declaration, instantiation and initialization  
  
        //printing array  
        for(int i=0;i<4;i++)  
            System.out.println(a[i]);  
  
    }  
}
```

Output:

```
33  
3  
4  
5
```



## Exercise Programs:

- Write a Java Program to print the elements of Java array using the for-each loop.

Output
Array Elements are:200 23 45 56

- Write a Java Program to print the largest number in an Array.

Input	Output
200 23 45 56	Largest Number=200

Write a Java Program to find the sum of all the elements in an Array.

Input	Output
200 23 45 56	Sum =324

Write a Java Program to print the smallest number in an Array.

Input	Output
200 23 45 56	Smallest Number=23

# Changing an Array Element

To change the value of a specific element, refer to the index number:

```
class Vehicle {  
    public static void main(String[] args) {  
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
        System.out.println(cars[0]);  
        cars[0] = "Opel";  
        System.out.println(cars[0]);  
    }  
}
```

```
Volvo  
Opel
```

## Passing Array as Parameters to Methods

Just like other objects, arrays can be passed as parameters to methods. The following method takes a int array as an argument and prints the data stored in each element of array:

```
import java.util.Scanner;
public class array {
    public int max(int [] array) {
        int max = 0;

        for(int i=0; i<array.length; i++ ) {
            if(array[i]>max) {
                max = array[i];
            }
        }
        return max;
    }
}
```

```

public int min(int [] array) {
    int min = array[0];
    for(int i = 0; i<array.length; i++ )
    {
        if(array[i]<min) {
            min = array[i];
        }
    }
    return min;
}

```

```

public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the array range");
    int size = sc.nextInt();
    int[] arr = new int[size];
    System.out.println("Enter the elements of the array ::");

    for(int i=0; i<size; i++) {
        arr[i] = sc.nextInt();
    }
    array m = new array();
    System.out.println("Maximum value in the array is::"+m.max(arr));
    System.out.println("Minimum value in the array is::"+m.min(arr));
}
}

```



# MultiDimensional Array

- A multidimensional array is an array of arrays.
- To create a two-dimensional array, add each array within its own set of **curly braces**:
- Syntax to declare 2dimensional Array(2D) in Java:
  - `dataType[][] arrayName;` (or)
  - `dataType [][] arrayName;` (or)
  - `dataType arrayName[][];` (or)

## Example1: 2D Array

```
int X[][]=new int[3][3];    //3 row and 3 column
```

	Column 0	Column 1	Column 2
Row 0	<b>x[0][0]</b>	<b>x[0][1]</b>	<b>x[0][2]</b>
Row 1	<b>x[1][0]</b>	<b>x[1][1]</b>	<b>x[1][2]</b>
Row 2	<b>x[2][0]</b>	<b>x[2][1]</b>	<b>x[2][2]</b>

```
x[0][0]=10;  
x[0][1]=20;  
x[0][2]=36;  
x[1][0]=47;  
x[1][1]=58;  
x[1][2]=69;  
x[2][0]=70;  
x[2][1]=8;  
x[2][2]=9;
```

10	20	36
47	58	69
70	8	9



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

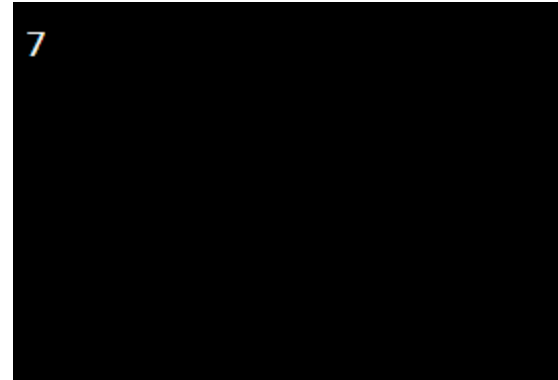


### Example:

```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```

- myNumbers is an 2D array.
- To access the elements of the myNumbers array, specify two indexes(row and column value)

```
class Main
{
    public static void main(String[] args)
    {
        int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7,9} };
        int x = myNumbers[1][2];
        System.out.println(x);
    }
}
```



We can also use a for loop inside another for loop to get the elements of a two-dimensional array (we still have to point to the two indexes):

### Example1:

```
class Example2
```

```
{
    public static void main(String[] args)
    {
        int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7,8} };
        for (int i = 0; i <2; ++i)
        {
            for(int j = 0; j <4 ; ++j)
            {
                System.out.println(myNumbers[i][j]);
            }
        }
    }
}
```

### OUTPUT

```
1
2
3
4
5
6
7
8
```



**PRESIDENCY  
UNIVERSITY**

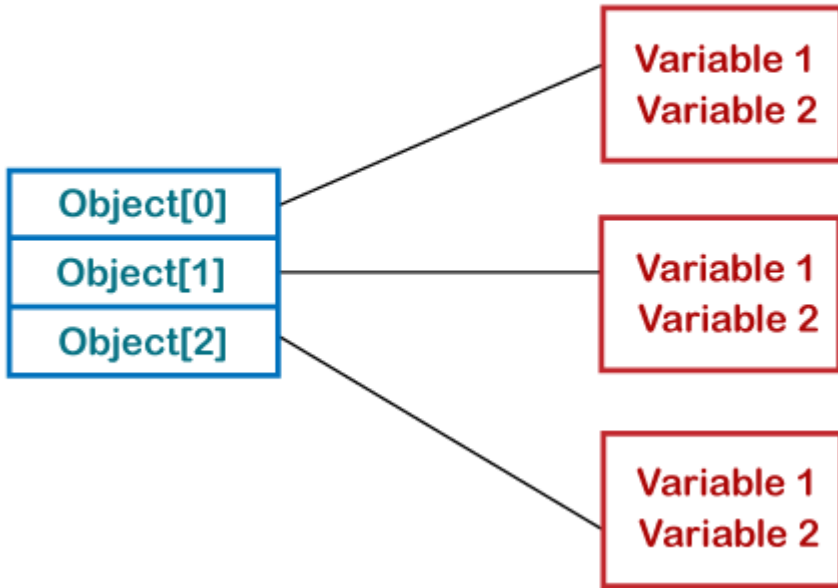
Private University Estd. in Karnataka State by Act No. 41 of 2013



# Array Of Objects

- Java is an object-oriented programming language. Most of the work done with the help of objects. We know that an array is a collection of the same data type that dynamically creates objects and can have elements of primitive types. Java allows us to store objects in an array.
- In Java, the class is also a user-defined data type. An array that contains class type elements are known as an array of objects. It stores the reference variable of the object.

## Arrays of Objects



### Creating an Array of Objects:

Before creating an array of objects, we must create an instance of the class by using the new keyword. We can use any of the following statements to create an array of objects.

### Syntax:

```
ClassName obj[]=new ClassName[array_length]; //declare and instantiate an array of objects
```



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



```
public class ArrayOfObjects
{
    public static void main(String args[])
    {
        //create an array of product object
        Product[] obj = new Product[5] ;
        //create & initialize actual product objects using constructor
        obj[0] = new Product(23907,"Dell Laptop");
        obj[1] = new Product(91240,"HP 630");
        obj[2] = new Product(29823,"LG OLED TV");
        obj[3] = new Product(11908,"MI Note Pro Max 9");
        obj[4] = new Product(43590,"Kingston USB");
```

```
//display the product object data
        System.out.println("Product Object 1:");
        obj[0].display();
        System.out.println("Product Object 2:");
        obj[1].display();
        System.out.println("Product Object 3:");
        obj[2].display();
        System.out.println("Product Object 4:");
        obj[3].display();
        System.out.println("Product Object 5:");
        obj[4].display();
    }
}
```

# Java Strings

- In Java, a string is a sequence of characters. For example, "hello" is a string containing a sequence of characters 'h', 'e', 'l', 'l', and 'o'.
- We use double quotes to represent a string in Java. For example,
  - `// create a string`
- String type = "Java programming";
- Here, we have created a string variable named type. The variable is initialized with the string Java Programming.



## Example: Create a String in Java

```
class Main {  
    public static void main(String[] args) {  
  
        // create strings  
        String first = "Java";  
        String second = "Python";  
        String third = "JavaScript";  
  
        // print strings  
        System.out.println(first);    // print Java  
        System.out.println(second);   // print Python  
        System.out.println(third);    // print JavaScript  
    }  
}
```



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# String Operations

- Java String provides various methods to perform different operations on strings. We will look into some of the commonly used string operations.
- To find the length of a string, we use the **length() method** of the String.

```
class Main {  
    public static void main(String[] args) {  
  
        // create a string  
        String greet = "Hello! World";  
        System.out.println("String: " + greet);  
  
        // get the length of greet  
        int length = greet.length();  
        System.out.println("Length: " + length);  
    }  
}
```

## Output

```
String: Hello! World  
Length: 12
```



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# concat()

```
class Main {  
    public static void main(String[] args) {  
  
        // create first string  
        String first = "Java ";  
        System.out.println("First String: " + first);  
  
        // create second  
        String second = "Programming";  
        System.out.println("Second String: " + second);  
  
        // join two strings  
        String joinedString = first.concat(second);  
        System.out.println("Joined String: " + joinedString);  
    }  
}
```

## Output

```
First String: Java  
Second String: Programming  
Joined String: Java Programming
```



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



## Methods of Java String:

Methods	Description
<a href="#"><u>contains()</u></a>	checks whether the string contains a substring
<a href="#"><u>substring()</u></a>	returns the substring of the string
<a href="#"><u>join()</u></a>	join the given strings using the delimiter
<a href="#"><u>replace()</u></a>	replaces the specified old character with the specified new character
<a href="#"><u>replaceAll()</u></a>	replaces all substrings matching the regex pattern
<a href="#"><u>replaceFirst()</u></a>	replace the first matching substring
<a href="#"><u>charAt()</u></a>	returns the character present in the specified location
<a href="#"><u>getBytes()</u></a>	converts the string to an array of bytes
<a href="#"><u>indexOf()</u></a>	returns the position of the specified character in the string
<a href="#"><u>compareTo()</u></a>	compares two strings in the dictionary order
<a href="#"><u>compareToIgnoreCase()</u></a>	compares two strings ignoring case differences
<a href="#"><u>trim()</u></a>	removes any leading and trailing whitespaces
<a href="#"><u>format()</u></a>	returns a formatted string
<a href="#"><u>split()</u></a>	breaks the string into an array of strings
<a href="#"><u>toLowerCase()</u></a>	converts the string to lowercase
<a href="#"><u>toUpperCase()</u></a>	converts the string to uppercase
<a href="#"><u>valueOf()</u></a>	returns the string representation of the specified argument
<a href="#"><u>toCharArray()</u></a>	converts the string to a char array
<a href="#"><u>matches()</u></a>	checks whether the string matches the given regex
<a href="#"><u>startsWith()</u></a>	checks if the string begins with the given string
<a href="#"><u>endsWith()</u></a>	checks if the string ends with the given string
<a href="#"><u>isEmpty()</u></a>	checks whether a string is empty or not
<a href="#"><u>intern()</u></a>	returns the canonical representation of the string
<a href="#"><u>contentEquals()</u></a>	checks whether the string is equal to charSequence
<a href="#"><u>hashCode()</u></a>	returns a hash code for the string
<a href="#"><u>subSequence()</u></a>	returns a subsequence from the string



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



## Creating strings using the new keyword:

So far we have created strings like primitive types in Java.

Since strings in Java are objects, we can create strings using the new keyword as well. For example,

**// create a string using the new keyword**

`String name = new String("Java String");`

In the above example, we have created a string name using the new keyword.

Here, when we create a string object, the `String()` constructor is invoked.

### Example: Create Java Strings using the new keyword

```
class Main {  
    public static void main(String[] args) {  
  
        // create a string using new  
        String name = new String("Java String");  
  
        System.out.println(name); // print Java String  
    }  
}
```



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



# String Buffer

- StringBuffer in java is used to create modifiable String objects. This means that we can use StringBuffer to append, reverse, replace, concatenate and manipulate Strings or sequence of characters. Corresponding methods under StringBuffer class are respectively created to adhere to these functions.
- In Java, Strings are known to be immutable or un-editable, unless overwritten upon. This is where StringBuffer class comes into picture, where it can accommodate character sequences and still enable a mutable feature.

# String Buffer Methods

## append() Method

➤ The append() method concatenates the given argument with this String.

```
class StringBufferExample
```

```
{
```

```
public static void main(String args[])
```

```
{
```

```
StringBuffer sb=new StringBuffer("Hello ");
```

```
sb.append("Java");           //now original string is changed
```

```
System.out.println(sb);      //prints Hello Java
```

```
}
```

```
}
```

## insert() Method:

The insert() method inserts the given String with this string at the given position.

```
class StringBufferExample2
{
    public static void main(String args[])
    {
        StringBuffer sb=new StringBuffer("Hello ");
        sb.insert(1,"Java");           //now original string is changed
        System.out.println(sb);       //prints HJavaello
    }
}
```



## replace() Method:

The replace() method replaces the given String from the specified beginIndex and endIndex.

```
class StringBufferExample3
{
public static void main(String args[])
{
StringBuffer sb=new StringBuffer("Hello");
sb.replace(1,3,"Java");
System.out.println(sb);      //prints HJavallo
}
}
```



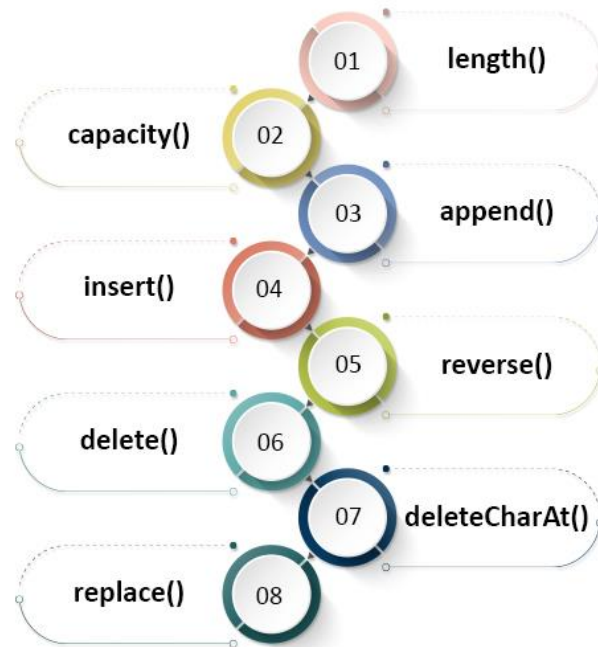
**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



## Methods of StringBuffer class

### StringBuffer Methods in Java



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013



## Test Your Coding Skills

1	Java Program to Count Repeated Elements in an Array
2	Java Program to Find Sum of Two Arrays in Java
3	Write a Java program to calculate the average of numbers or find the average of elements of the given array.
4	Write a Java program to print the following grid. Expected Output : ----- ----- ----- -----
5	Write a Java program to find the index of an array element.



**PRESIDENCY  
UNIVERSITY**

Private University Estd. in Karnataka State by Act No. 41 of 2013

