Topic: “Strings in Java”

Strings In Java

- What is a String?

❑ String is a sequence of characters represented in double quotes(“”).

❑ The Java platform provides the String class to create and manipulate strings

❑String objects are immutable!

✓ - That means once a string object is created it cannot be altered. For mutable string, one can use **StringBuffer** and **StringBuilder** classes.

✓ - Normally objects in java are created using **new** keyword e.g.

```
String name;
Name= new String(“abcd”);
```

OR

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

✓However String objects can also be created "implicitly"

```
String name;
Name = "abcd";
```

✓The String class is defined in Java.lang package.
✓To use String as mutable, use StringBuffer class.

Dynamic Initialization of Strings:

```
BufferedReader br = new BufferedReader( new InputStreamReader(System.in));
        String city = br.readLine();

Scanner sc = new Scanner(System.in);
        String state = sc.nextLine();
        String state1 = sc.next();
```

String Concatenation:

- Java String can be concatenated using ‘+’ operator.

```
          String firstName = “name”;
          String lastName = “last”;
System.out.println(firstName + “  ” + lastName);
```

String of Arrays:

- An array of String can also be created..

```
String cities [] = new String[5];
```

- Which will create an array of Cities of size 5 o hold String constants.

String Indexes:

The 12 characters in the String “Java is fun” have indexes 0 to 11.

String	J	a	v	a		i	s		f	u	n	.
Index	0	1	2	3	4	5	6	7	8	9	10	11

String Methods:

- The String class contains many useful methods for string - processing applications.

- A String method is called by writing String object, a dot, the name of the method and a pair of parentheses to enclose any arguments.

- If a String method returns a value, then it can be placed anywhere that a value of its type can be used…

```
String greeting = “Hello”;
int count = greeting.length();
System.out.println(“Length is “ + count);
```

- Always count from zero when referring to the position or index of a character in s String.

- charAt(). Returns the character at the specific index(position).

- compareTo(). Compares two Strings lexicographically.

- concat(). Append a String to the end of another String.

- contains(). Checks whether a String contains a sequence of characters.

- equals(). Compares two Strings. Return true if the Strings are equal , and false if not.

- indexOf(). Returns the position of the first found occurrence of specified characters in a String.