



Republic of the Philippines
Department of Education
REGION III
SCHOOLS DIVISION OFFICE OF NUEVA ECija

LEARNING ACTIVITY SHEET
SPECIAL PROGRAM IN ICT 9
BASIC PROGRAMMING 9
Fourth Quarter, Week 2

Name of Learner: _____ Date: _____
Grade Level /Section: _____

STRING OPERATIONS

Background Information for Learners

Java String provides various methods to perform different operations on strings. We will look into some of the commonly used string operations.

1. Get Length of a String

To find the length of a string, we use the `length()` method of the String. For example,

```
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
```

In the above example, the `length()` method calculates the total number of characters in the string and returns it.

2. Join two Strings or the String Concatenation

We can join two strings in Java using the `concat()` method. For example,

```
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
```

In the above example, we have created two strings named **first** and **second**. Notice the statement, `String joinedString = first.concat(second);`

Here, the `concat()` method joins `first` and `second` and assigns it to the `joinedString` variable.

Using + Operator for Concatenation

In Java, you can also use the `+` operator to concatenate two strings. For example,

```
class Main {
    public static void main(String[] args) {
        String str1 = "Learn ";
        String str2 = "Java";

        // concatenate str1 and str2
        System.out.println(str1 + str2); // "Learn Java"

        // concatenate str2 and str1
        System.out.println(str2 + str1); // "JavaLearn "
    }
}
```

concat() Vs the + Operator for Concatenation

concat()	the + Operator
Suppose, str1 is <code>null</code> and str2 is <code>"Java"</code> . Then, <code>str1.concat(str2)</code> throws NullPointerException .	Suppose, str1 is <code>null</code> and str2 is <code>"Java"</code> . Then, <code>str1 + str2</code> gives "nullJava" .
You can only pass a String to the <code>concat()</code> method.	If one of the operands is a string and another is a non-string value. The non-string value is internally converted to a string before concatenation. For example, <code>"Java" + 5</code> gives <code>"Java5"</code> .

3. Compare two Strings

In Java, we can make comparisons between two strings using the `equals()` method. For example,

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

        // create 3 strings
        String first = "java programming";
        String second = "java programming";
        String third = "python programming";

        // compare first and second strings
        boolean result1 = first.equals(second);
        System.out.println("Strings first and second are equal: " + result1);

        // compare first and third strings
        boolean result2 = first.equals(third);
        System.out.println("Strings first and third are equal: " + result2);
    }
}
```

Output

```
Strings first and second are equal: true
Strings first and third are equal: false
```

In the above example, we have created 3 strings named **first**, **second**, and **third**. Here, we are using the `equal()` method to check if one string is equal to another. The `equals()` method checks the content of strings while comparing them.

Learning Competency with Code

Identify basic string operations

Apply string operations in future lessons

Exercises/Activities

ACTIVITY 1

Direction: Using the length () method, create a code that will calculates the total number of characters in the string “I love programming”. Write your code in a one whole sheet of paper.

ACTIVITY 2

Direction: Using the + operator for string concatenation, create a code that will join your first name and last name by declaring 2 different strings. Write your code in a one whole sheet of paper.

ACTIVITY 3

Direction: Using the equals() method, create a code that will compare 2 different strings “Passed” and “Failed”. Write your code in a one whole sheet of paper.

Scoring Rubrics

	20	15	10
Accuracy	The code is 100% accurate and follows the correct loop structure being asked. The code is running and the output was met.	The code has 2-5 error/incorrect code	The code has 5 or more error/incorrect code

Reflection

In a one whole sheet of paper explain each string operations discussed.

References for Learners

<https://www.programiz.com/java-programming/library/string/concat>

<https://www.programiz.com/java-programming/string>

Prepared by: **NOEMI F. MAGNO**
Name of Writer

Noted by: **LABERNE A. LADIGNON, JR**
Division ICT Coordinator/ OIC EPS