

# Strings in Java

## Assignment Questions

1. Write a simple String program to take input from user

Ans 1.

```
package Strings;
```

```
class Pwjava {
```

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

```
{
```

```
    String sca = " ";
```

```
    Scanner scan = new Scanner(System.in);
```

```
    String b = scan.next();
```

```
System.out.println(b);
```

```
}
```

```
}
```

2. How do you concatenate two Strings in Java? Give an example?

Ans 2. In Java, We can concatenate two Strings using the '+' operator. Here's an example:

```
String firstString = "Hello, ";
```

```
String secondString = "world! ";
```

```
String concatenatedString = firstString + secondString;
```

```
System.out.println(concatenatedString); // Output: Hello, world!
```

In this example, the '+' operator combines the contents of 'firstString' and 'secondString' to create a new String 'concatenatedString'.

3. How do you find the length of a string in Java Explain with an example?

Ans 3. In Java, we can find the length of a string using the 'length()' method of the 'string' class.

Here's an example:

```

public class StringLengthExample {

    public static void main(String[] args) {

        String str = "Hello, world!";

        int length = str.length();

        System.out.println("Length of the string: " + length);

    }

}

```

In this example, the 'length()' method is called on the 'str' variable, which contains the string "Hello, world!". The returned value is the length of the string, which is 13 in this case. The program then prints out the length using the 'System.out.println()' method.

4. How do you compare two strings in Java? Give an example?

Ans 4. In Java, we can compare two strings using '==' operator or the 'equals()' method. The '==' operator checks if two string references point to the same memory location, while the 'equals()' method compares the actual content of the strings. Here's an example:

```

String str1 = "Hello";

String str2 = new String("Hello");

String str3 = "Hello";

boolean usingEqualOperator = str1 == str2; // Returns false

boolean usingEqualsMethod = str1.equals(str2); // Returns true

boolean usingEqualOperatorWithSameReference = str1 == str3; // Returns true

```

```
boolean usingEqualsMehtodWithSameContent = str1.equals(str3); // Returns true
```

In the example above, ‘usingEqualOperator’ is ‘false’ because ‘str1’ and ‘str2’ refer to different memory locations even though they have the same content.

‘usingEqualsMethod’ is ‘true’ because the ‘equals( )’ method compares the content and finds that they are the same.

Similarly,

‘usingEqualOperatorWithSameReference

‘and

‘usingEqualsMethodWithSameContent’

are both ‘true’ because both ‘str1’ and ‘str3’ refer to the same memory location and have the same content.

5. Write a program to find the length of the string “refrigerator” ?

Ans 5.

```
public class StringLength {  
    public static void main(String[] args) {  
        String word = “refrigerator”;  
        int length = word.length();  
        System.out.println(“The length of the string is: “ + length);  
    }  
}
```

}

6. Write a program to check if the letter 'e' is present in the word "Umbrella" .

Ans 6.

```
public class CheckLetter {  
    public static void main(String[] args) {  
        String word = "Umbrella";  
        char letterToCheck = 'e';  
        boolean isPresent = false;  
  
        for (int i = 0; i < word.length( ); i++) {  
            if (word.charAt(i) == letterToCheck) {  
                isPresent = true;  
                break;  
            }  
        }  
  
        if (isPresent) {  
            System.out.println("The letter 'e' is present in the word.");  
        } else {  
            System.out.println("The letter 'e' is not present in the word.");  
        }  
    }  
}
```

7. Write a program to delete all the constants from the string “Hello,have a good day”.

Ans 7.

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

        String input = “Hello, have a good day”;

        String vowels = “aeiouAEIOU”; // Assuming these are the vowels

        StringBuider result = new StringBuilder( );

        for (char c : input.toCharArray() ) {

            if (Character.isWhitespace(c) || vowels.indexOf(c) != -1) {

                result.append(c);

            }

        }

        String output = result.toString( ) ;

        System.out.println(output);

    }

}
```