Southern Luzon State University
College of Engineering
Computer Engineering Department

CPE13 Object Oriented Programming

Activity 3: Character and Strings

Name: REIMARC G. CORPUZ
Section: BSCPE III-GF

Date: 10 - 4 - 2022
Score:_____

1.1 Introduction

Array is a collection of similar data type items which are used to store group of data of same data type. Arrays can be of any data type (e.g. string, character, integer and float) and must have constant size. Array index always starts with 0 and uses continuous memory locations to store data in array.

String is an array of character and Java platform provides the String class to create and manipulate strings. String can be created by simply writing: String str1 = “Hello Java”; or using another String object: String str2 = new String(str1);.

1.2 Objective

- To use Java programming language to create a program that exhibits different uses of String and Characters.
- To conceptualize the process and manipulate the program
- To distinguish different String class and functions.

1.3 Problem

Write a program that determines the length, concatenation, Switching from Lower Case to Upper Case and vice versa, Reversion. List of method string.length(), “+” sign for concatenation, string.toLowerCase(), string.toUpperCase(), and string.reverse().

The output of the Program: Screenshot

1. 2 String input

2. determine the length of the first string input

3. concatenate first and second string

4. first string will be in Upper Case

5. second string will be in Lower Case

6. second string will be reversed

Note: both string input must have atleast 1 Upper / Lower Case letter.

1.4 Follow up Questions:

1. How do you implement each String method?

I implement each string method by defining the string to be input. I identify and analyze the number of characters within that string and then do what is needed in the output. I also used some methods or syntax to make the code simple and useful and display the desired output.

2. Does String methods useful? How?

Yes, because not all the data of the user is not only numbers or maybe a single letter. There is some system that needs a specific word to compile in their database. So that the system can be understandable and specific data to what is needed.

3. What practical applications can they be applied?

It is useful for data gathering that needs the names or any string data of the respondent. By displaying the text in the output, the user of the system will easily understand what to do or what is asked in the system compared to numbers there is a pattern that the user needs to follow and to understand.

1.5 Conclusion

After coding the different methods of string, I concluded that there is some other way to display the same output but different syntax. For example, in concatenation, I used .concat() but also I can use the operation + between the two variable inputs of the string. Meanwhile, when it comes to the use of the string I understand that string ensures the input data to what is the preferred format and the value of each property is as expected. Also, because the string is a combination of characters, in creating another method I can use its single characters to create a condition to display like for example in a reverse form of the input string.

Code of the Program:

```
import java.util.Scanner;

public class CORPUZ3 {



	public static void main( String args[] )

	{

		Scanner input = new Scanner(System.in);

		System.out.print("Enter the first string: ");

		String string1 = input.nextLine();

		System.out.print("Enter the second string: ");

		String string2 = input.nextLine();



		int length1 = string1.length();

		System.out.print("\nThe length of the first
string is: " + length1);

		int length2 = string2.length();

		System.out.print("\nThe length of the second
string is: " + length2);



		String concatenate = string1.concat(" ");

		concatenate = concatenate.concat(string2);

		System.out.print("\n\nConcatenate strings: "
+ concatenate);



		System.out.print("\n\n" + string1 + " to: " +
string1.toUpperCase());

		System.out.print("\n" + string2 + " to: " +
string2.toUpperCase());



		System.out.print("\n\n" + string1 + " to: " +
string1.toLowerCase());

		System.out.print("\n" + string2 + " to: " +
string2.toLowerCase() + "\n\n");



		System.out.print(string2 + " to: ");

		char[] reverse = string2.toCharArray();

		for (int i = reverse.length - 1; i >= 0; i--)

		System.out.print(reverse[i]);

	}



}
```

DISPLAY OUTPUT