

Object Oriented Programming Lab

Lab Manual – Cstrings

Exercise 1:

- i. Run following piece of code and paste the output in space given below:

```
#include<iostream>
using namespace std;

void main()
{
    //Exercise 1
    char charArray[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd' };
    cout << charArray << endl;
}
```

Output:

Why is it printing garbage at the end?

- ii. Run following piece of code and paste the output in space given below:

```
char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };
cout << myStr1 << endl;
```

Output:

What is the difference between two arrays declared above? Which one is a c-string?

- iii. **[StringLenght]** Write a function **int StringLenght(char*)** that takes a cstring and returns its length. Remember that length of c-string does not include null character. Length of myStr1 should print 11.
- iv. Comment the declaration of myStr1 and declare it as the code segment given below. Run and test your program now.

```
//char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };  
char myStr1[] = "Hello World";
```

- v. Comment the declaration of myStr1 and declare it as the code segment given below. Run and test your program now.

```
//char myStr1[] = { 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '\0' };  
//char myStr1[] = "Hello World";  
char myStr1[20] = "Hello World";
```

- vi. Run the code given below and check its behavior:

```
char myStr1[5] = "Helo";  
cout << "Initial String:\t" << myStr1 << endl;  
  
int len = 0;  
for (; myStr1[len] != '\0'; len++);  
cout << "Lenght of myStr1:\t" << len << endl << endl << endl;  
  
cout << "Enter another string of size 4 :";  
cin >> myStr1;  
  
cout << "String entered by the user is:" << myStr1 << endl;  
  
for (len = 0 ; myStr1[len] != '\0'; len++);  
cout << "Lenght of myStr1:\t" << len << endl << endl << endl;  
  
cout << "Enter another string of size 5 or Greater :";  
cin >> myStr1;  
  
cout << "String entered by the user is:" << myStr1 << endl;  
  
for (len = 0 ; myStr1[len] != '\0'; len++);  
cout << "Lenght of myStr1:\t" << len << endl << endl << endl;  
cout << "Program is going to terminate.\n";
```

Why did the program crash?

In the exercises given below, dynamically allocate char array of size 80 to save data. Make sure your program does not cause any memory leakage.

Exercise 2: [GetCharacterCount] Write a function `int GetCharacterCount(char* myString, char c)` that takes a character *ch* and a c-string *myString* and returns total number of occurrences of *ch* in *myString*.

Sample Output:

```
myString: Pakistan
ch: a
Total No of Occurances: 2
```

Exercise 3: Update the program written in Exercise 7 to accept a sentence in *myString* and test it. You need to replace `"cin>>myString;"` with `"cin.getline(myString,50);"`

Sample Output:

```
myString: I am Pakistani
ch: a
Total No of Occurances: 3
```

Practice Problems

- Write a program that takes a c-string *myStr* and two characters *charToFind* and *charToReplace* from user and replaces all the occurrences of *charToFind* with *charToReplace* in *myStr*. Your program should create a space of 50 characters on heap in order to save *myStr*.

Sample output:

```
InputString: ddsdfhgrtsdfhjghjksdd
CharToFind: d
CharToReplace: $
ModifiedString: $$$sfhgrts$fhjghjks$$
```

- Write a program that takes a character *ch* and a CString *myStr* from user and removes all the occurrences of *ch* from *myStr*.

Sample Output:

myStr: cabccdefcfdcxzcc

ch: 'c'

Modified String: abdeffd xyz

TrimStart(char* str)

Write a function that takes a string and removes all the space in start of the string.

Sample Output:

Before TrimStart

str: " Hello How are you?"

After TrimStart

str: "Hello How are you?"

TrimEnd(char* str)

Write a function that takes a string and removes all the space at the end of the string.

Sample Output:

Before Trim End

str: "I love programmin g. "

After Trim End

str: "I love programmin g."

Hint: Traverse the array from end

END