

Fundamentals of Programming

Lecture 10

Chamila Karunatilake

Department of Information and Communication Technology

Faculty of Technology

University of Sri Jayewardenepura

chamilakarunatilake@sjp.ac.lk

Strings



C Strings

- In programming, it's often needed storing words, phrases and sentences.
- C provides **character strings** to facilitate this.
- Strings are actually one-dimensional array of characters

H	E	L	L	o
---	---	---	---	---

H	o	w		a	r	e		y	o	u	?
---	---	---	--	---	---	---	--	---	---	---	---

C Strings

- There is a small difference between a raw character array and a character string.
- Character string is terminated by a **null** character '\0'.

H	E	L	L	o	\0
---	---	---	---	---	----

H	o	w		a	r	e		y	o	u	?	\0
---	---	---	--	---	---	---	--	---	---	---	---	----

String Initialization

- There are two ways to initialize strings.
- The string can be initialized by giving the elements of the string with in braces. The null character is added at the end.

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

- The string can be given with in double quotation marks. Here, the null character or the size is not needed.

```
char greeting[] = "Hello";
```

Output String using printf()

- String can be printed using printf() function one character at a time as normal array.
- However, C provides %s format specifier for string
- The whole string can be printed at once.

```
char name[] = "Naymar" ;  
printf("%s", name) ;  
printf("Your name is %s", name) ;
```

Input String using scanf()

- Using the same format specifier, %s, a string input can be received from standard input(keyboard).

```
char name[20];
```

```
scanf("%s", name);
```

```
printf("Your name is %s.", name);
```

- However, this will only take a string up to a white space. If the string we input is “Kane Williamson” it will only takes Kane.
- As a solution to that, we could use special format specification instead of “%s”.

Input String using scanf()

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```


Input String using scanf()

```
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%[^\\t\\n]", name);
    printf("Your name is %s.", name);
    return 0;
}
```

String Manipulation

- C supports a wide range of functions that manipulate strings
- To use them, `string.h` standard library should be included at the top of the program.

`#include <string.h>`

- We discuss few important string manipulation functions.

String Manipulation

Function	Description
<code>strlen(s1)</code>	Returns the length of string s1.
<code>strcpy(s1,s2)</code>	Copies string s2 into string s1.
<code>strcat(s1,s2)</code>	Concatenates string s2 onto the end of string s1.
<code>strcmp(s1, s2)</code>	Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
<code>strlwr(s1)</code>	Converts string s1 to lowercase.
<code>strupr(s1)</code>	Converts string s2 to uppercase.

String Manipulation

```
include <string.h>

int main () {

    char str1[12] = "Hello";
    char str2[12] = "World";
    char str3[12];
```

String Manipulation

```
int len = strlen(str1);  
printf("strlen(str1) : %d\n", len );  
  
int b = strcmp(str1, str2);  
printf("strcmp( str1, str2) : %d\n", b );  
  
strcpy(str3, str1);  
printf("strcpy( str3, str1) : %s\n", str3 );
```

String Manipulation

```
    strcat( str1, str2);  
    printf("strcat( str1, str2):    %s\n", str1 );  
  
    strlwr(str1);  
    printf("strlwr(str1) :    %s\n", str1 );  
  
   strupr(str1);  
    printf("strupr(str1) :    %s\n", str1 );  
    return 0;  
}
```

gets() and puts()

- Functions gets() and puts() are two string functions to take string input from the user and display it respectively.

```
gets (name) ;  
puts (name) ;
```

gets() and puts()

```
#include <stdio.h>
int main()
{
    char name[30];
    printf("Enter name: ");
    gets(name);
    printf("Name: ");
    puts(name);
    return 0;
}
```


Palindrome

- A **palindrome** is a word, number, phrase, or other sequence of characters which reads the same backward as forward.

madam racecar abba rotator

Malayalam

1783871 1551



- According to Guinness World Records, the Finnish 19 letter word **saippuakivikauppias** (a soapstone vendor), is the world's longest palindromic word in everyday use

Program to Check Palindrome

```
#include <stdio.h>
int main()
{
    char inputStr[30];
    printf("Enter any String: ");
    gets(inputStr);
    int i = 0;
    int j = strlen(inputStr) - 1;
```

```
while (i < j)
{
    if (inputStr[i++] != inputStr[j--])
    {
        printf("%s is Not Palindrome", inputStr);
        return;
    }
}
printf("%s is palindrome", inputStr);
return 0;
}
```

Count the total number of words in a string

```
#include <stdio.h>
int main()
{
    char inputStr[30];
    printf("Enter a Sentence: ");
    gets(inputStr);

    int i = 0;
    int wrd = 1;
```

```
while (inputStr[i] != '\0')
{
    if (inputStr[i] == ' ' || inputStr[i] == '\n' ||
        inputStr[i] == '\t')
    {
        wrd++;
    }
    i++;
}
printf("Number of words : %d\n", wrd);
return 0;
}
```

Questions?
