**Department of Electrical and Computer Engineering, NSU**

**CSE 115L: Fundamentals of Computer Programming (Section 4)**

**Lab 14 (Strings), Faculty: Rsl**

**Strings:** Strings are actually one-dimensional array of characters terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null

|  |  |
| --- | --- |
| **Example** | **Declaration & Initialization of strings** |
| "c string"  Here, "c string" is a string. When, compiler encounters strings, it appends null character at the end of string   |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | C |  | s | t | r | i | n | g | \0 | | Strings are declared in C in similar manner as arrays. Only difference is that, strings are of char type:  char s[5];  In C strings can be initialized in many ways:  char c[ ]="abcd";  OR,  char c[5]="abcd";  OR,  char c[ ]={'a','b','c','d','\0'};  OR;  char c[5]={'a','b','c','d','\0'};   |  |  |  |  |  | | --- | --- | --- | --- | --- | | a | b | c | d | \0 | |

**(String declaration, input and output)**

|  |  |
| --- | --- |
| **Example 1** | **Example 2 ( input using loop)** |
| #include<stdio.h>  int main()  {  int i;  char str[6]={'H','E','L','L','O','\0'};  char st[]="world";  for(i=0; i<6; i++){  printf("%c",str[i]);  }  // printf("%s \n",str);  printf("%s \n",st);  return 0;  } | #include<stdio.h>  int main()  {  char str[10];  int i;  for(i=0; i<5; i++)  {  fflush(stdin);  printf("Enter character:");  scanf("%c",&str[i]);  }  printf("%s",str);  return 0;  } |

|  |  |
| --- | --- |
| **Example 3: alternative way to take input** | **Example 4: strlen(str ), strcat(str1,str2) & strcpy(str1,str2) function in C** |
| #include<stdio.h>  int main()  {  char name[20];  scanf("%s", name);  printf("%s", name);  // gets(name);  // the above gets takes name with space in between  // puts(name);  return 0;  } | #include<stdio.h>  #include<string.h>  int main()  {  char str1[10],str2[10],str3[20];  int len;  printf("Enter String 1:");  gets(str1);  printf("Enter String 2:");  gets(str2);  len=strlen(str1);  printf("The length of the string 1 is: %d\n", len);  strcat(str1,str2);  printf("%s\n",str1);  strcpy(str3,str1);  printf("%s",str3);  return 0;  } |

|  |  |
| --- | --- |
| **Example 5: passing string as functions argument** |  |
| #include<stdio.h>  void printString(char s[ ]);  int main()  {  char str[10];  printf("Enter Strings:");  gets(str);  printString(str);  return 0;  } | void printString(char s[ ])  {  int i=0;  while(s[i]!='\0')  {  printf("%c",s[i]);  i++;  }  } |

|  |  |
| --- | --- |
| **Example 6: Finding particular character from string** |  |
| #include<stdio.h>  void findChar(char s[],char a);  int main()  {  char str[10],ch;  printf("Enter Strings:");  gets(str);  printf("Enter character to find:");  scanf("%c",&ch);  findChar(str,ch);  return 0;  } | void findChar(char s[],char a)  {  int i=0;  while(s[i]!='\0')  {  if(s[i]==a){  printf("Found character %c at index: %d",a,i);  break;  }  i++;  }  if(i==strlen(s))  {  printf("Character not found!\n");  }  } |