



Strings in C++ Computer Programming

Content S

- Strings (*LL 02*)
- Strings in C++ (*LL 02*)
- C-String (*LL 04*)
- Declaring C-String (*LL 04*)
- Initializing C-String (*LL 04*)
- Outputting C-String (*LL 04*)
- Accessing characters of C-String (*LL 04*)
- Inputting C-String (*LL 04*)

*LL 02 = Learning Level 02 –
Comprehension,*

*LL 04 = Learning Level 04 –
Analysis*



Content

S

- Functions in <string.h> <cstring> header files (*LL 04*)
- Functions in <ctype.h> <cctype> header files (*LL 04*)
- String Object (*LL 04*)
- Declaring String Object (*LL 04*)
- Initializing String Object (*LL 04*)
- Concatenating String Objects (*LL 04*)
- Copying String Objects (*LL 04*)
- Accessing characters of String Object (*LL 04*)
- Inputting String Object (*LL 04*)

LL 02 = Learning Level 02 – Comprehension, LL 04 = Learning Level 04 – Analysis

Content S

- Length of String Object (*LL 04*)
- Comparing two String Objects (*LL 04*)
- Substring of String Object (*LL 04*)
- Finding string in String Object (*LL 04*)
- Replacing string in String Object (*LL 04*)
- Program Examples (*LL 04*)

*LL 02 = Learning Level 02 –
Comprehension,*

*LL 04 = Learning Level 04 –
Analysis*

String

S

- String is the collection/sequence of characters.
- Generally it is combination of two or more characters.
- In programming string is the collection of 0 or more characters.
- The string with 0 character is considered as the empty string.
- There is no any built-in data type for storing and processing strings in C++.
- In C++ every string literal is enclosed in double quotes.

String

- Following are the examples of string literals:

"IQRA University" "CS"	//String with multiple words
""	//String with single word
" "	//Empty string
"c"	//String with single character

Strings in C++

- There are two ways to create strings in C++:



C-Strings



String Objects

C-String

- C-String is also referred to as **Character-String**.
- It is the method to create strings in C language hence also called as **C-Style** string.
- The C-String is actually a one-dimensional array of characters which is terminated by a **null** character **'\0'**.
- Thus a null-terminated string contains the characters that comprise the string followed by a null.

C-String can be considered as special character array. It is actually a character array but terminated by null character.



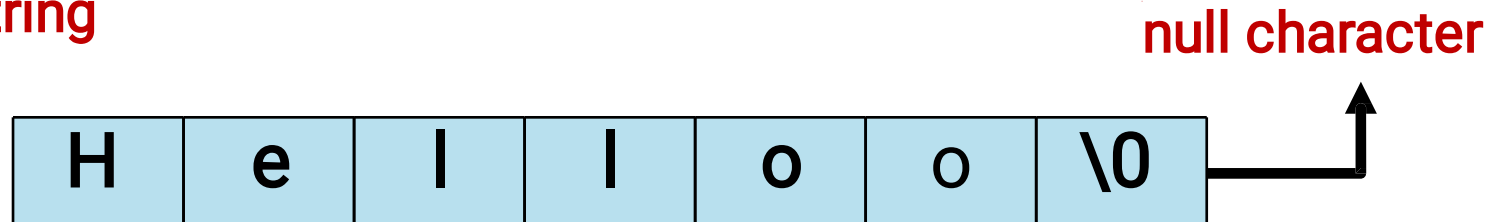
C-String

Consider the followings:

- This is Character Array



- This is C-String



Declaring C-String

- A C-String can be declared as:

```
char name [10];
```

```
char address [50];
```

```
char rollno [12];
```



Initializing C-String

- A C-String can be initialized in one of the following ways:

```
char name [] = "Asghar" ;
```

```
char name [10] = "Asghar" ;
```

```
char name [] = { 'A' , 's' , 'g' , 'h' , 'a' , 'r' , '\0' } ;
```

```
char name [7] = { 'A' , 's' , 'g' , 'h' , 'a' , 'r' , '\0' } ;
```

Outputting C-String

- A C-String can be displayed just by using its name in cout statement.

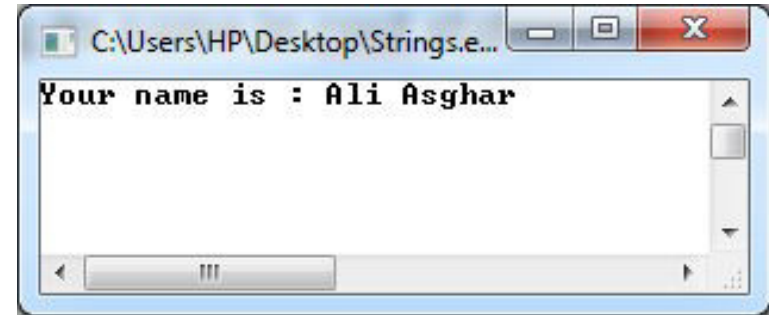
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    char name[20] = "Ali Asghar";

    cout<<"Your name is : "<<name;

    getch();
    return 0;
}
```



Accessing characters of C-String

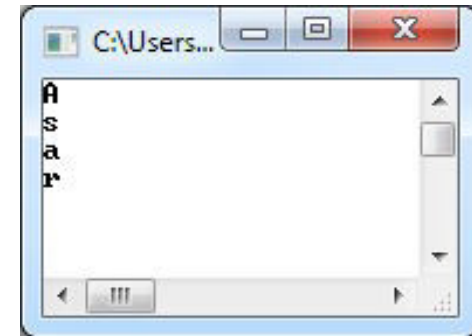
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    char name[20] = "Ali Asghar";

    cout<<name[0]<<endl;
    cout<<name[5]<<endl;
    cout<<name[8]<<endl;
    cout<<name[9]<<endl;

    getch();
    return 0;
}
```



Inputting C-String

- A C-String can be inputted by using cin statement.

cin>>str	Inputs single word only, cannot accept multiple words or string with spaces. <i>str</i> is the name of the string.
cin.get(str, MAX)	Inputs single line of text, cannot accept multiple lines of text. <i>str</i> is the name of the string and <i>MAX</i> is the maximum characters in <i>str</i> .
cin.get(str, MAX, Delimiter)	Inputs multiple lines of text. <i>str</i> is the name of the string, <i>MAX</i> is the maximum characters in <i>str</i> and <i>Delimiter</i> is the character which delimits the string of multiple lines.



Inputting C-String

Using (`cin>>str`)

```
#include<iostream>
#include<conio.h>

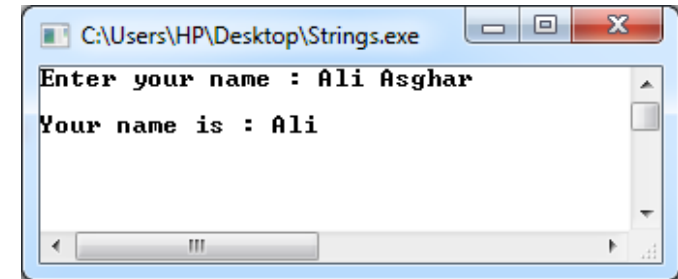
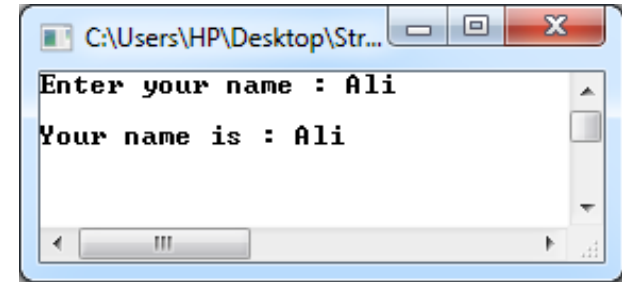
using namespace std;

int main()
{
    char name[20];

    cout<<"Enter your name : ";
    cin>>name;

    cout<<endl<<"Your name is : "<<name;

    getch();
    return 0;
}
```



Inputting C-String

Using (~~cin.get(str, MAX)~~)

```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    char name[20];

    cout<<"Enter your name : ";
    cin.get(name,20);

    cout<<endl<<"Your name is : "<<name;

    getch();
    return 0;
}
```


Inputting C-String

Using (`cin.get(str, MAX, Delimiter)`)

```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    char text[100];

    cout<<"Enter your text : ";
    cin.get(text, 100, '$');

    cout<<endl<<"Your entered : "<<text;

    getch();
    return 0;
}
```

Functions in <string.h> <cstring> header files

• <string.h> and <cstring> header files contain few useful functions that can perform operations on c-strings.

Function	Purpose
strcpy(str1, str2)	Copies string str2 in to str1.
strcat(str1, str2)	Concatenates/joins str1 and str2. The result will be stored in str1.
strlen(str)	Returns the length of the string str.
strcmp(str1, str2)	Compares whether str1 is larger, smaller or equals to str2.



strcpy(str1, str2)

- It copies c-string str2 in to str1.

```
#include<iostream>
#include<conio.h>
#include<cstring>

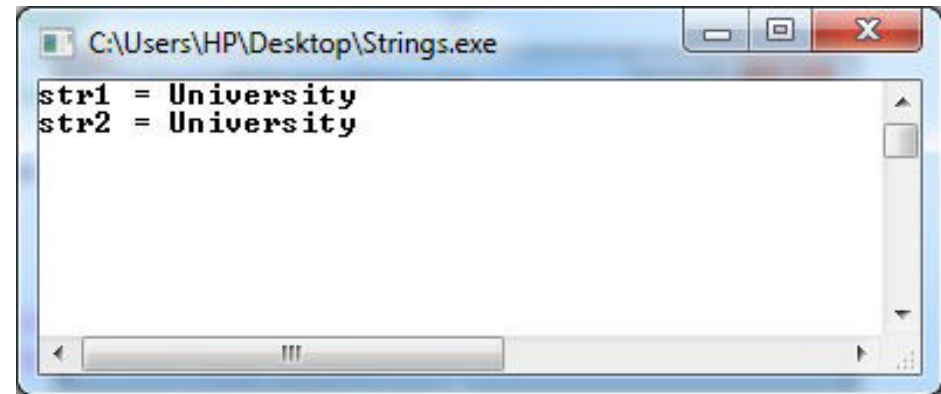
using namespace std;

int main()
{
    char str1[50] = "mehran";
    char str2[50] = "University";

    strcpy(str1, str2);

    cout<<"str1 = "<<str1<<endl<<"str2 = "<<str2;

    getch();
    return 0;
}
```



strcat(str1, str2)

- It concatenates c-string str1 in to

```
#include<iostream>
#include<conio.h>
#include<cstring>

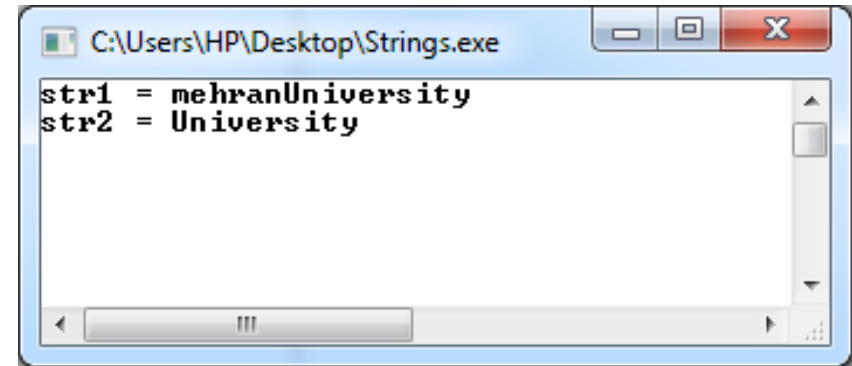
using namespace std;

int main()
{
    char str1[50] = "mehran";
    char str2[50] = "University";

    strcat(str1, str2);

    cout<<"str1 = "<<str1<<endl<<"str2 = "<<str2;

    getch();
    return 0;
}
```



strlen(str)

- It returns length of str.

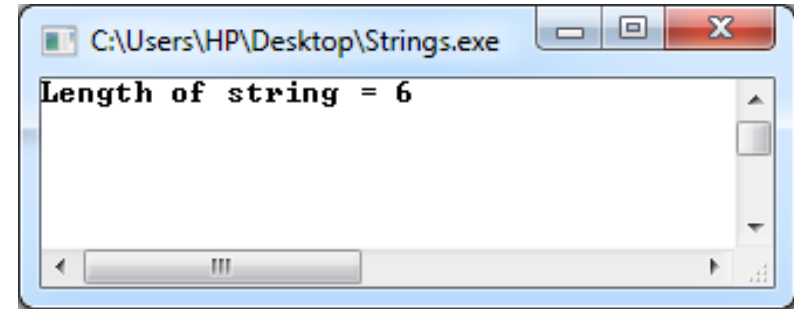
```
#include<iostream>
#include<conio.h>
#include&ltcstring>

using namespace std;

int main()
{
    char str1[50] = "mehran";

    cout<<"Length of string = "<<strlen(str1);

    getch();
    return 0;
}
```



strcmp(str1, str2)

- It compares str1 and str2.
- It will return either of three values: (< 0 , 0 , > 0)
- If it returns < 0 then str1 is smaller than str2.
- If it returns > 0 then str1 is larger than str2.
- If it returns 0 then str1 and str2 are equal.

strcmp(str1, str2)

```
#include<iostream>
#include<conio.h>
#include<cstring>

using namespace std;

int main()
{
    char name1[20];
    char name2[20];

    cout<<"Enter name1 : ";
    cin.get(name1,20);
    cin.sync();
    cout<<"Enter name2 : ";
    cin.get(name2,20);

    if(strcmp(name1,name2) < 0)
        cout<<"name1 is smaller than name2";
    else if(strcmp(name1,name2) > 0)
        cout<<"name1 is larger than name2";
    else
        cout<<"name1 and name2 are equal";

    getch();
    return 0;
}
```

Functions in <ctype.h> <cctype> header

• <ctype.h> and <cctype> header files contain few useful functions that perform operations on characters.

Function	Purpose
isalpha(char)	Checks if a character is an alphabetic.
isalnum(char)	Checks if a character is an alphanumeric.
islower(char)	Checks if a character is a lower case character.
isupper(char)	Checks if a character is a upper case character.
isdigit(char)	Checks if a character is a digit.
isspace(char)	Checks if a character is a space character.
ispunct(char)	Checks if a character is a punctuation character.
toupper(char)	Coverts a character to uppercase.
tolower(char)	Coverts a character to lowercase.



Functions in <ctype.h> <cctype> header files

```
#include<iostream>
#include<conio.h>
#include<cctype>

using namespace std;

int main()
{
    if(isalpha('A'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(isalpha('?'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(isalnum('5'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;
```

```
    if(isalnum('g'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(islower('R'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(islower('e'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(isupper('W'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;

    if(isupper('w'))
        cout<<"Yes"<<endl;
    else
        cout<<"No"<<endl;
```

Functions in <ctype.h> <cctype> header files

```
if(isspace('t'))
    cout<<"Yes"<<endl;
else
    cout<<"No"<<endl;

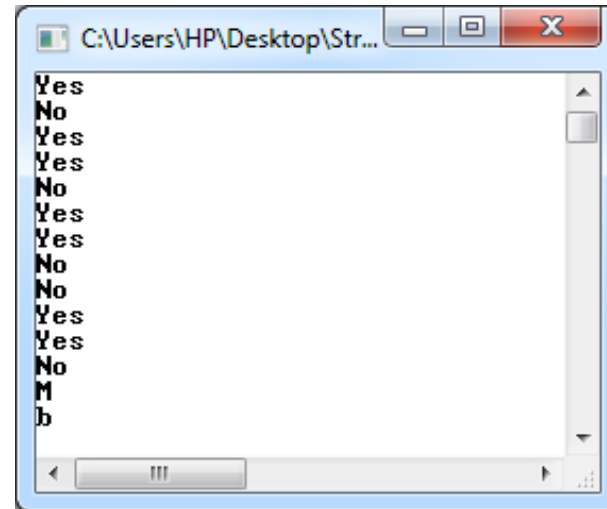
if(isspace(' '))
    cout<<"Yes"<<endl;
else
    cout<<"No"<<endl;

if(ispunct('?'))
    cout<<"Yes"<<endl;
else
    cout<<"No"<<endl;

if(ispunct('k'))
    cout<<"Yes"<<endl;
else
    cout<<"No"<<endl;
```

```
cout<<(char) (toupper('m'))<<endl;
cout<<(char) (tolower('B'))<<endl;

getch();
return 0;
}
```



String Object

- String object is another way to create strings in C++.
- String object is the **object** of class **string**.
- The size of string objects can be varied during the execution of the program unlike the C-Strings.
- The syntax of string objects is more intuitive than C-Strings.
- More functions are available to perform operations on string objects as compared to C-Strings.



Declaring String Object

- String object can be declared as:

```
string name ;
```

```
string address ;
```

```
string rollno ;
```



Initializing String Object

- String object can be initialized in one of the following ways:

```
string name1 = "saira" ;
```

```
string name2 ( "saira" );
```

```
string name3 ( name2 );
```



Concatenating String Objects

```
#include<iostream>
#include<conio.h>

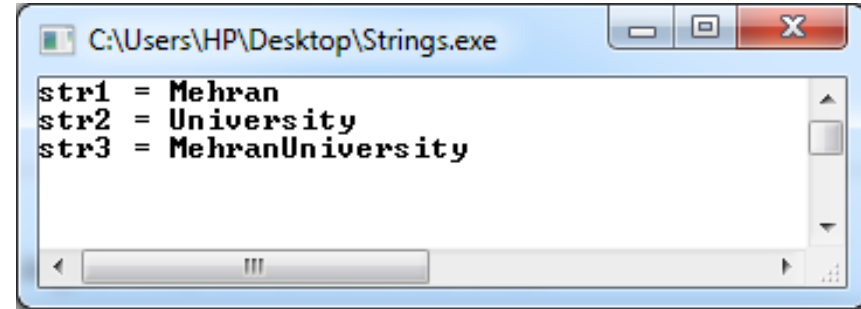
using namespace std;

int main()
{
    string str1 = "Mehran";
    string str2 = "University";
    string str3;

    str3 = str1 + str2;

    cout<<"str1 = "<<str1<<endl<<"str2 = "<<str2<<endl<<"str3 = "<<str3;

    getch();
    return 0;
}
```



Copying String Objects

```
#include<iostream>
#include<conio.h>

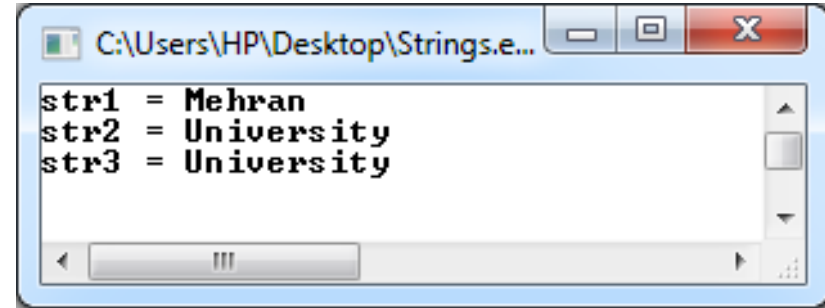
using namespace std;

int main()
{
    string str1 = "Mehran";
    string str2 = "University";
    string str3;

    str3 = str2;

    cout<<"str1 = "<<str1<<endl<<"str2 = "<<str2<<endl<<"str3 = "<<str3;

    getch();
    return 0;
}
```



Accessing characters of String Object

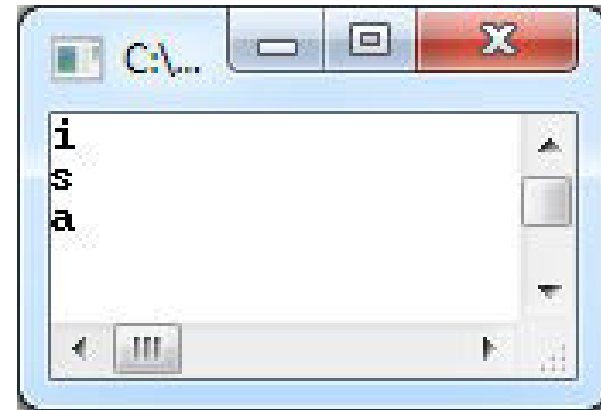
```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    string name = "Ali Asghar";

    cout<<name[2]<<endl;
    cout<<name[5]<<endl;
    cout<<name.at(8)<<endl;

    getch();
    return 0;
}
```



Inputting String Object

cin>>str	Inputs single word only, cannot accept multiple words or string with spaces. <i>str</i> is the name of the string.
getline(cin, str)	Inputs single line of text, cannot accept multiple lines of text. <i>str</i> is the name of the string.
getline(cin, str, Delimiter)	Inputs multiple lines of text. <i>str</i> is the name of the string and <i>Delimiter</i> is the character which delimits the string of multiple lines.

Inputting String

Using (`cin>>str`) Object

```
#include<iostream>
#include<conio.h>

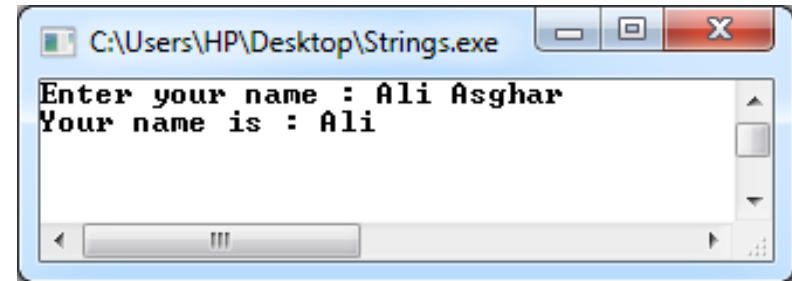
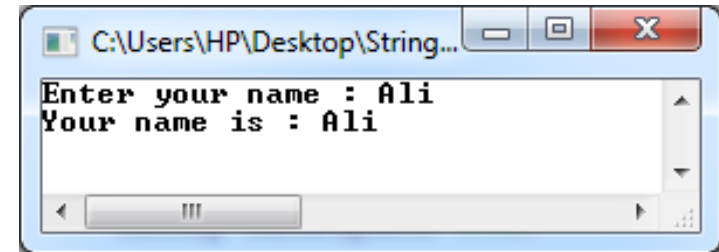
using namespace std;

int main()
{
    string name;

    cout<<"Enter your name : ";
    cin>>name;

    cout<<"Your name is : "<<name;

    getch();
    return 0;
}
```



Inputting String

Using (`getline(cin, str)`)
Object

```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    string name;

    cout<<"Enter your name : ";
    getline(cin, name);

    cout<<"Your name is : "<<name;

    getch();
    return 0;
}
```



Inputting String

Using (`getline(cin, str, Delimiter)`)
Object

```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    string text;

    cout<<"Enter your name : ";
    getline(cin, text, '$');

    cout<<endl<<"You entered "<<endl<<text;

    getch();
    return 0;
}
```



Length of String Object

```
#include<iostream>
#include<conio.h>

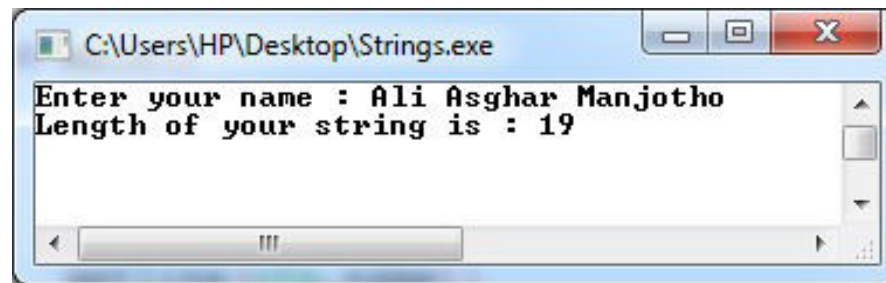
using namespace std;

int main()
{
    string name;

    cout<<"Enter your name : ";
    getline(cin,name);

    cout<<"Length of your string is : "<<name.length();

    getch();
    return 0;
}
```



Comparing two String Objects

```
#include<iostream>
#include<conio.h>

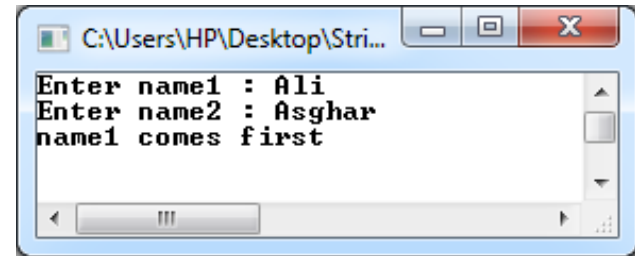
using namespace std;

int main()
{
    string name1;
    string name2;

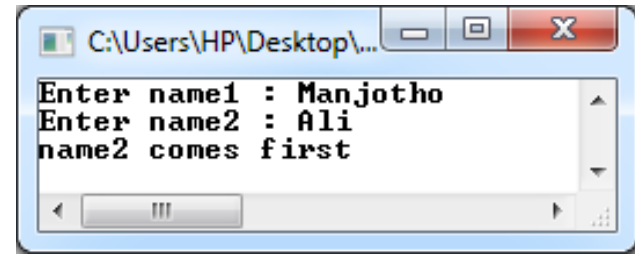
    cout<<"Enter name1 : ";
    cin>>name1;
    cout<<"Enter name2 : ";
    cin>>name2;

    if(name1 < name2)
        cout<<"name1 comes first";
    else if(name1 > name2)
        cout<<"name2 comes first";
    else
        cout<<"name1 and name2 are equal";

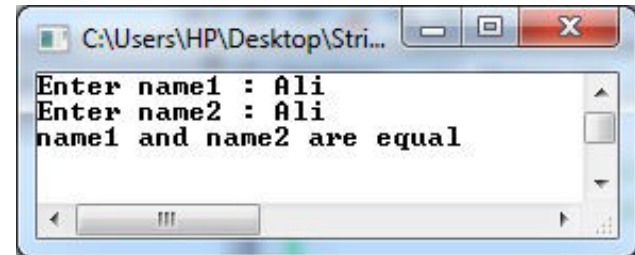
    getch();
    return 0;
}
```



```
C:\Users\HP\Desktop\Stri...
Enter name1 : Ali
Enter name2 : Asghar
name1 comes first
```



```
C:\Users\HP\Desktop\...
Enter name1 : Manjotho
Enter name2 : Ali
name2 comes first
```



```
C:\Users\HP\Desktop\Stri...
Enter name1 : Ali
Enter name2 : Ali
name1 and name2 are equal
```



Program Examples

Strings

Program Example 01

Problem Statement:

Write a program in C++ that inputs text from the user and converts all the uppercase letter in to lowercase and lowercase letters in to uppercase letters.



Program Example

```
( #include<iostream>
  #include<conio.h>
  #include<cstring>
  #include<cctype>

  using namespace std;

  int main()
  {
      char text[100];

      cout<<"Enter your text : ";
      cin.get(text,100,'$');

      for(int i=0; i<strlen(text); i++)
      {
          if(isupper(text[i]))
              text[i] = tolower(text[i]);
          else if(islower(text[i]))
              text[i] = toupper(text[i]);
      }

      cout<<endl<<text;

      getch();
      return 0;
  }
```

Program Example 02

Problem Statement:

Write a program in C++ that inputs text from the user and converts it in uppercase.



Program Example

```
#include<iostream>
#include<conio.h>

using namespace std;

int main()
{
    string text;

    cout<<"Enter your text : ";
    getline(cin,text,'$');

    for(int i=0; i<text.length(); i++)
    {
        text[i] = toupper(text[i]);
    }

    cout<<endl<<text;

    getch();
    return 0;
}
```

Program Example 02

