String Handling

Instructor: Isra Naz

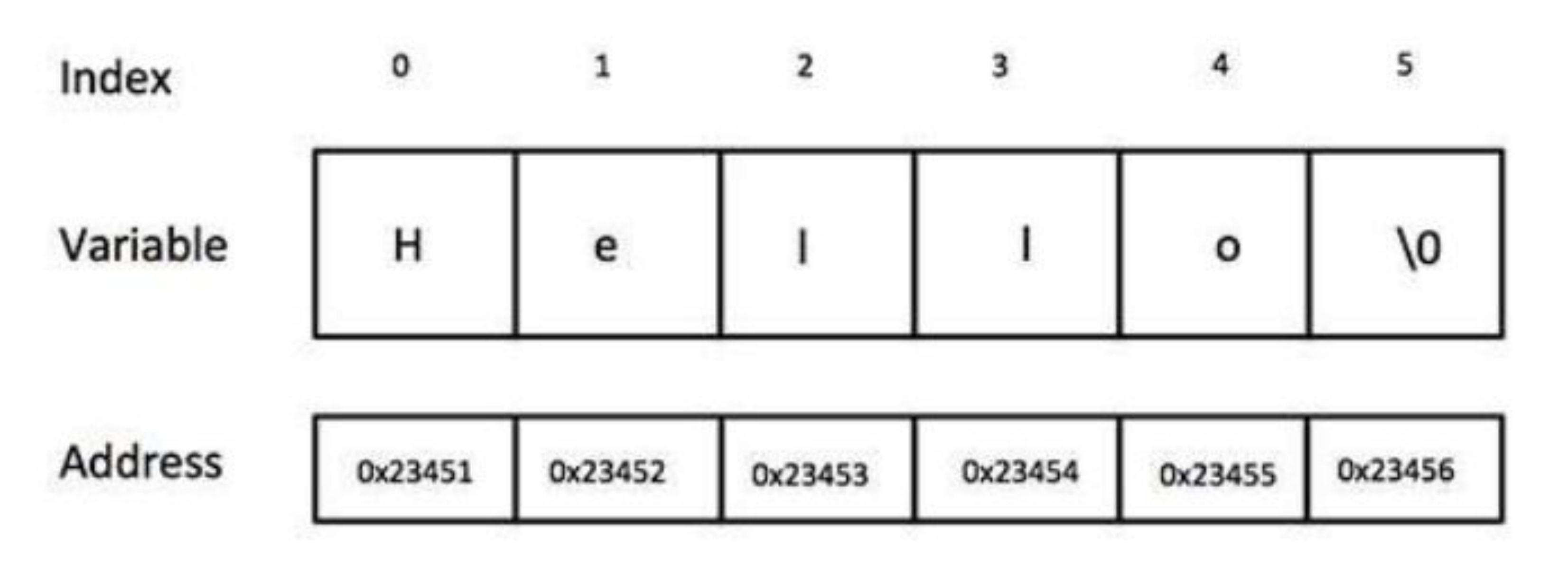
String

- A collection of characters written in double quotations is called string or string constant.
- It may consist of an alphabetic characters, digits and special symbols.
- A sting is stored as an array of characters and terminated by a special symbol known as null character.
- It is denoted by the escape sequence \0 and is used to indicate the end of string.

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

String

- The values stored in an array of characters can be manipulated individually using the index of the array.
- The user can input, process and displays the individual characters of string in the same way as an array.



Index	0	1	2	3	4	5
Variable	H	e	l	l	o	\0
Address	0x23451	0x23452	0x23453	0x23454	0x23455	0x23456

String Declaration

- C++ stores a string as an array of characters
- An array is a group of contiguous memory location that can store same type of data.
- **Syntax:**

```
char array_name[length];
```

Example

```
char book[20];
```

String Initialization

- **Syntax:**

char array_name[length]=value;

Example

char book[20] = “Object Oriented Programming”;

A sting can also be declared without indicating the length as follows

char name[] = “Pakistan”;

The above statement automatically declares a sting according to the length of value.

P	a	k	i	s	t	a	n	\0

Null Character

String Input

- A string value can be input from the user using different functions.
- C++ provides the following functions for string input
  - **cin**
  - **cin.getline()**
  - **cin.get()**

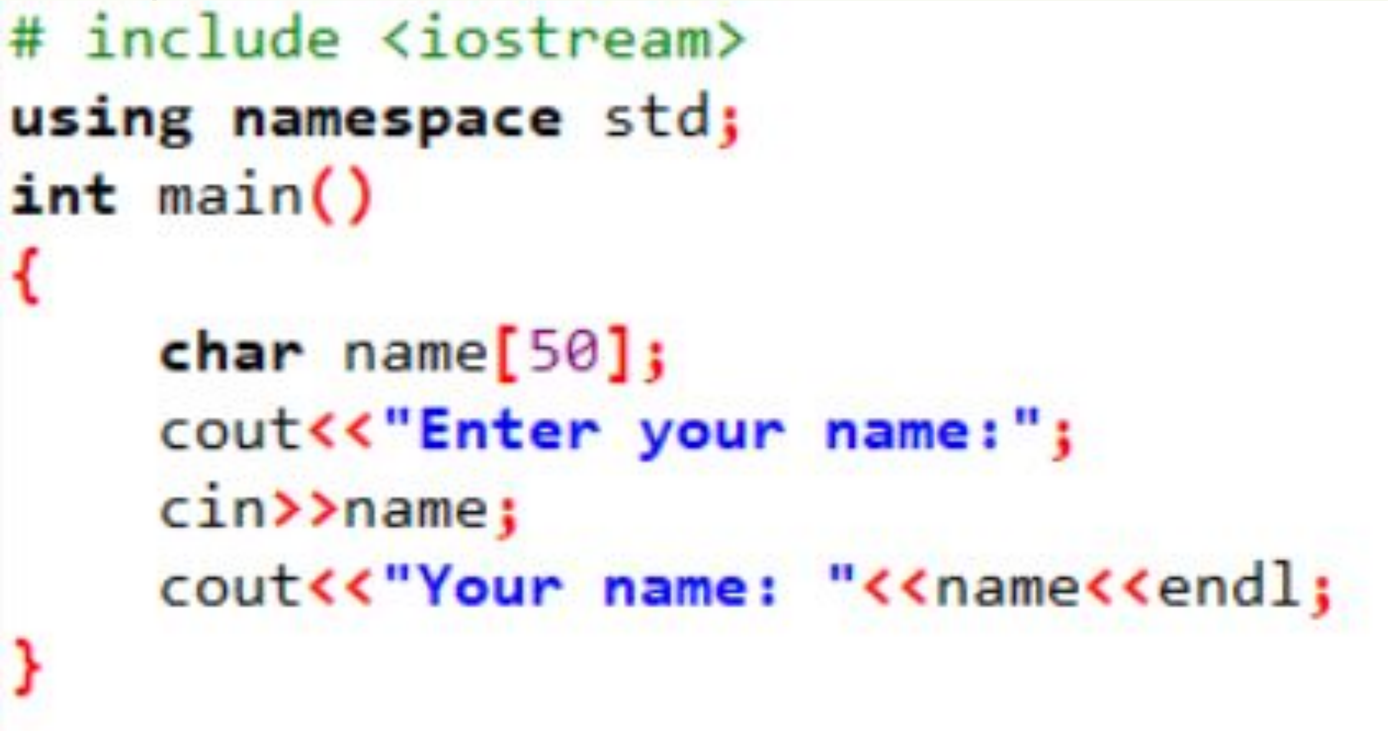
‘cin’ Object

- It is used to input a string value without any blank space in it.
- It does not support a sting with spaces.
- If a user inputs a string value that contains any space, the value before the space will be used.
- The remaining sting values will be ignored.

- **Syntax**
- cin>>str;

Example

- A program that inputs the name of the user using, cin



```
# include <iostream>
using namespace std;
int main()
{
    char name[50];
    cout<<"Enter your name:";
    cin>>name;
    cout<<"Your name: "<<name<<endl;
}
```

cin.getline()

- The getline() function of cin object is used to input any string value including blank spaces.
- The user can input any type of string value.
- **Syntax**

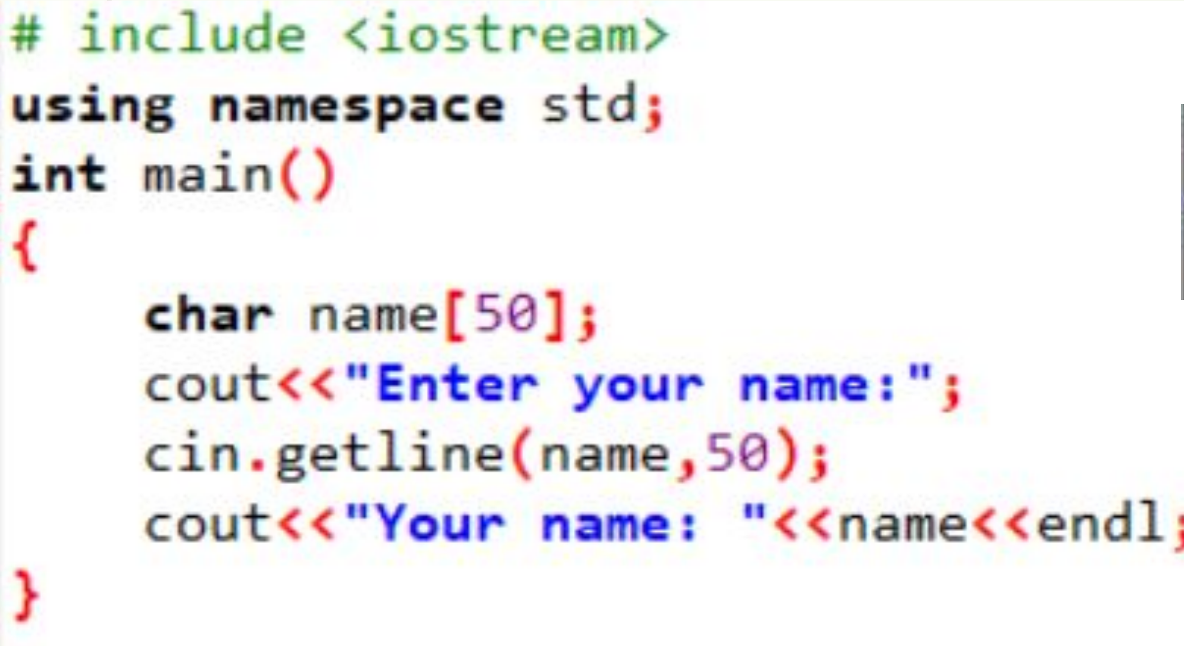
cin.getline(str, len);

str It indicates the name of string variables in which the value is to be stored.

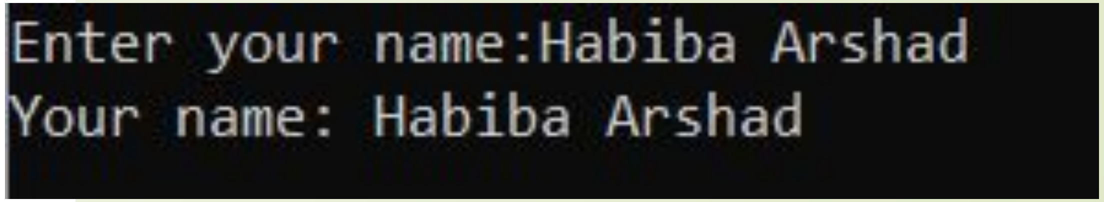
lenIt indicates the length of string variable.

Example

- A program that inputs the name of the user using, cin.getline() object and displays it.



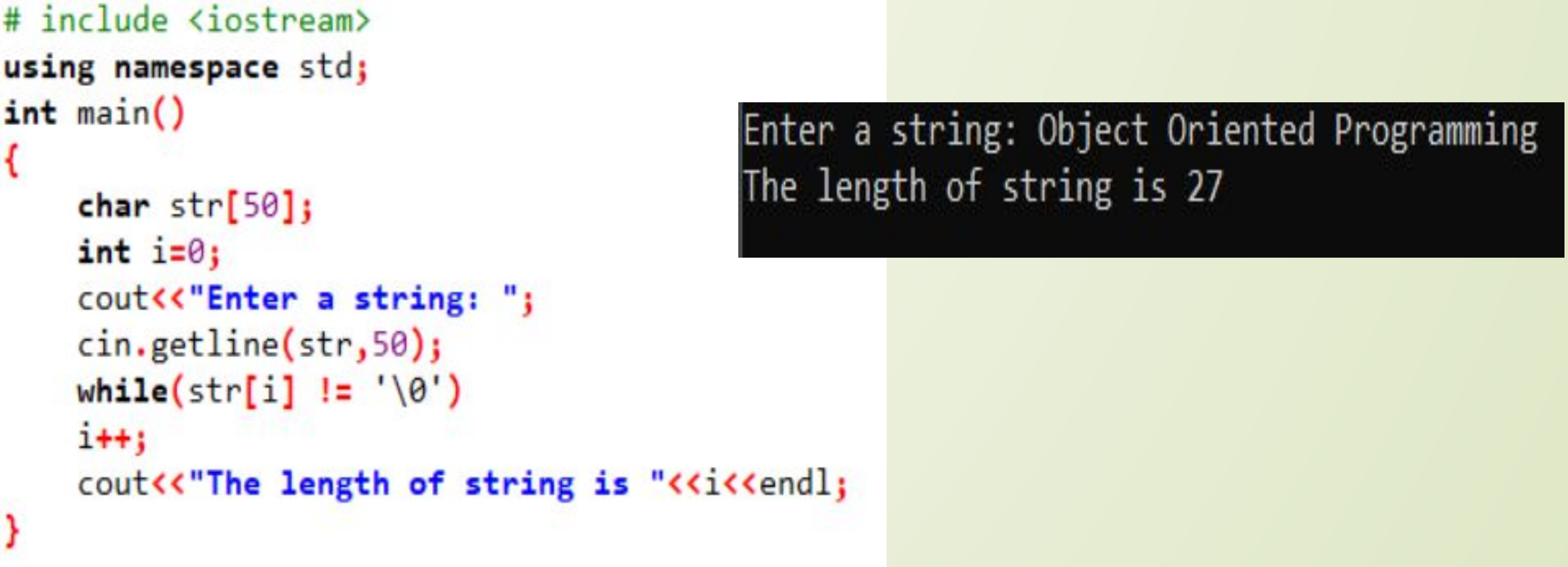
```
# include <iostream>
using namespace std;
int main()
{
    char name[50];
    cout<<"Enter your name:";
    cin.getline(name,50);
    cout<<"Your name: "<<name<<endl;
}
```



```
Enter your name:Habiba Arshad
Your name: Habiba Arshad
```

Example

- **Write a program that inputs a sting from the user and displays its length**



```
# include <iostream>
using namespace std;
int main()
{
    char str[50];
    int i=0;
    cout<<"Enter a string: ";
    cin.getline(str,50);
    while(str[i] != '\0')
    i++;
    cout<<"The length of string is "<<i<<endl;
}
```

```
Enter a string: Object Oriented Programming
The length of string is 27
```

cin.get()

- The get() function of cin object is used to input a single character.
- **Syntax**

```
cin.get(ch);
```

ch It indicates the name of character variable in which the value is to be stored.

Array of Strings

- An array of string is actually a two dimensional array of characters.
- Each row of the array represents one string.
- Each character in an array of strings is stored in a separate index of two dimensional array.
- **Syntax**

```
char str[rows][cols];
```

Example

```
char names[3][5];
```

Initializing Array of Strings

- An array of stings can be initialized in different ways
  - It can be initialized by assigning individual characters to each index in the array
  - It can also be initialized by assigning complete strings to each row in the array.

Example

```
    char names[3][5] = {‘a’ , ’b’ , ’c’ , ‘d’ . ‘e’
                        ‘f’ , ‘g’ , ‘h’ , ‘I’ , ‘j’
                         ‘k’ , ‘l’ , ‘m’ , ‘n’ , ‘o’}
```

Initializing Array of Strings

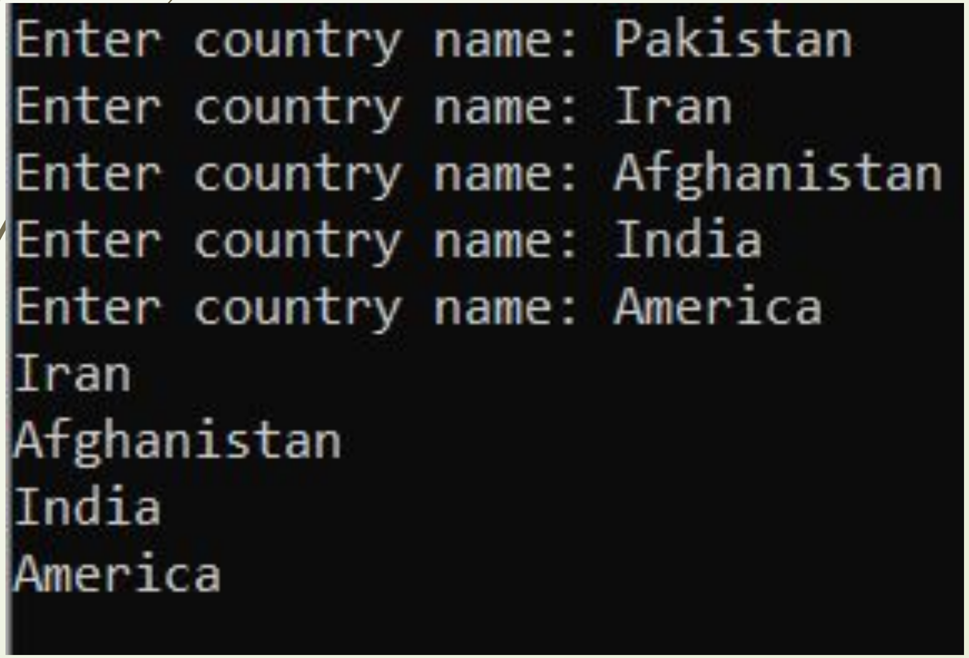
- **Example**
-
  ```
  char str[3][10] = {“Ali” ,
                     “Abdullah”,
                      “Usman”};
  ```

Example

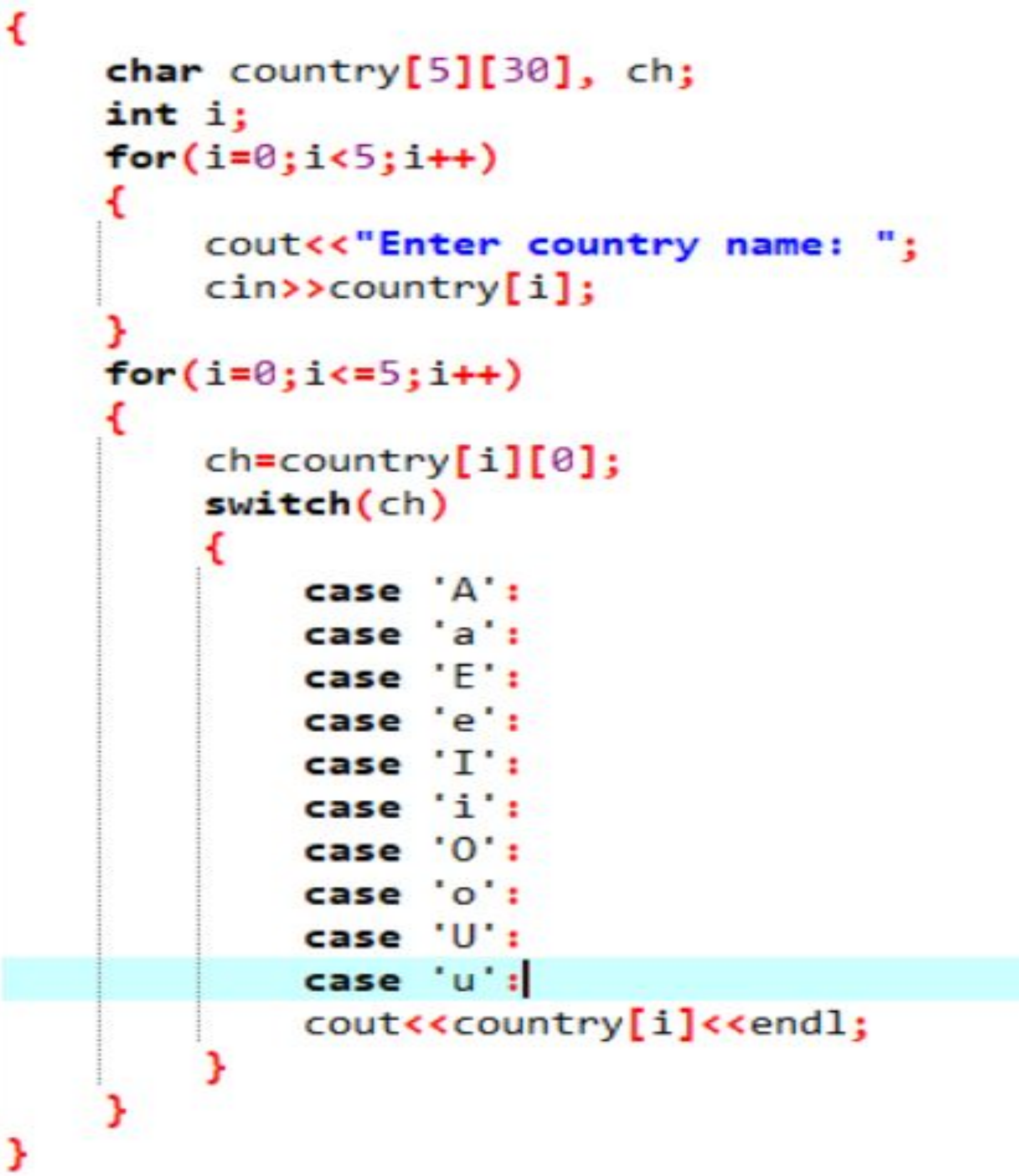
- Write a program that inputs the names of five countries. It only displays the countries name starts with a vowel.

Example

- Write a program that inputs the names of five countries. It only displays the countries name starts with a vowel.

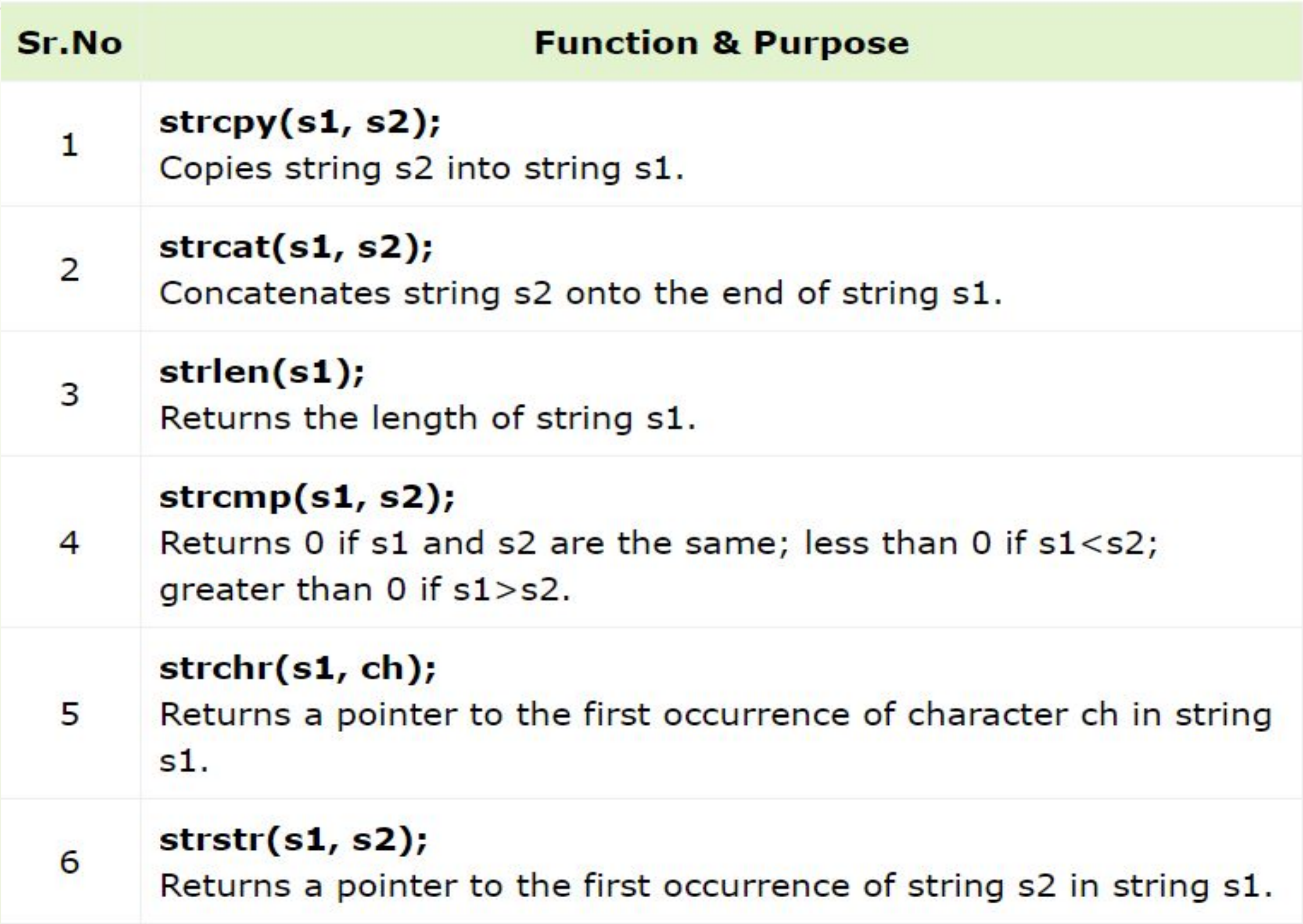


```
Enter country name: Pakistan
Enter country name: Iran
Enter country name: Afghanistan
Enter country name: India
Enter country name: America
Iran
Afghanistan
India
America
```



```
{
    char country[5][30], ch;
    int i;
    for(i=0;i<5;i++)
    {
        cout<<"Enter country name: ";
        cin>>country[i];
    }
    for(i=0;i<=5;i++)
    {
        ch=country[i][0];
        switch(ch)
        {
            case 'A':
            case 'a':
            case 'E':
            case 'e':
            case 'I':
            case 'i':
            case 'O':
            case 'o':
            case 'U':
            case 'u':
            cout<<country[i]<<endl;
        }
    }
}
```

String Functions (string.h)

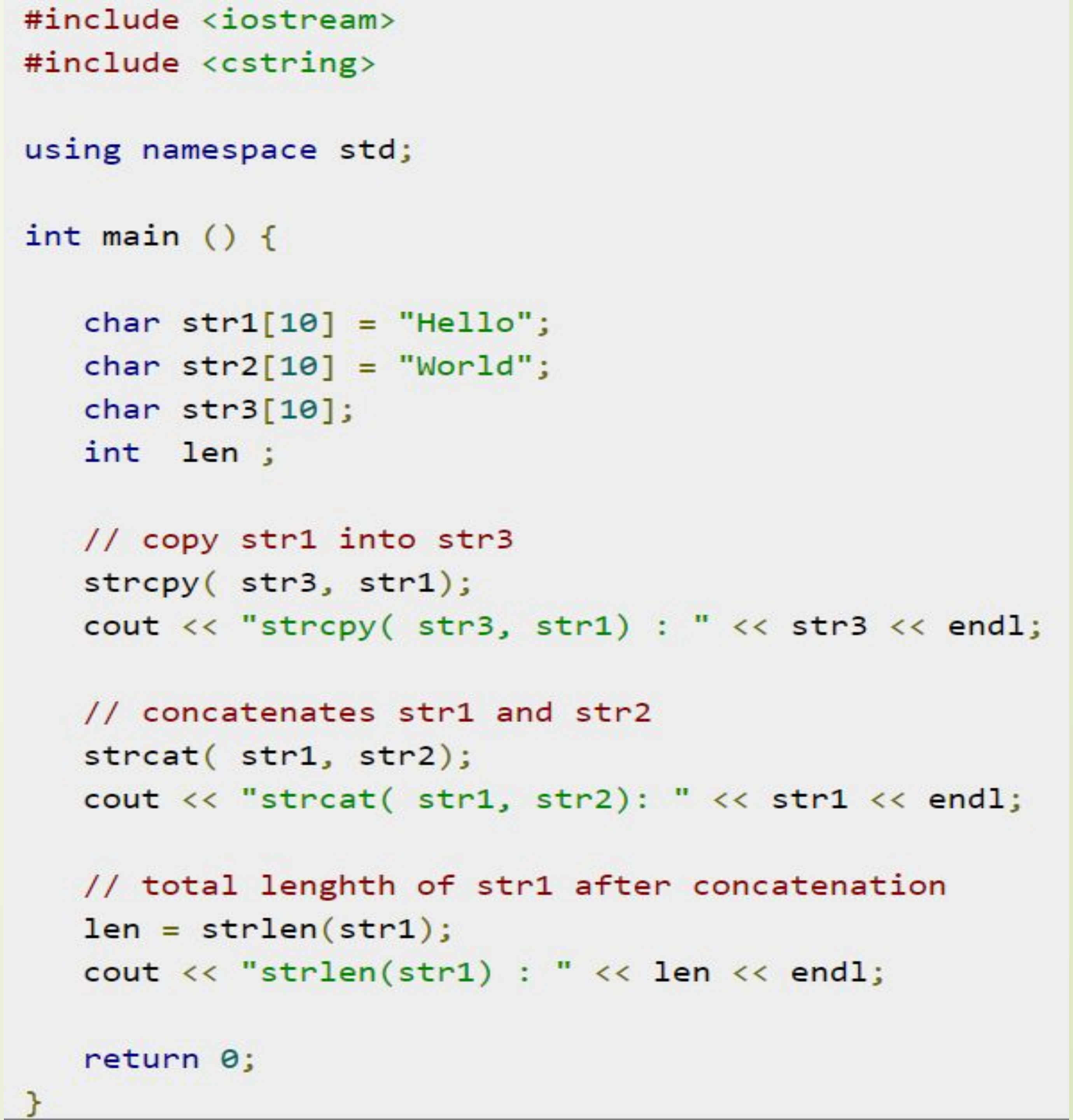


Sr.No	Function & Purpose
1	**strcpy(s1, s2);** Copies string s2 into string s1.
2	**strcat(s1, s2);** Concatenates string s2 onto the end of string s1.
3	**strlen(s1);** Returns the length of string s1.
4	**strcmp(s1, s2);** Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.
5	**strchr(s1, ch);** Returns a pointer to the first occurrence of character ch in string s1.
6	**strstr(s1, s2);** Returns a pointer to the first occurrence of string s2 in string s1.

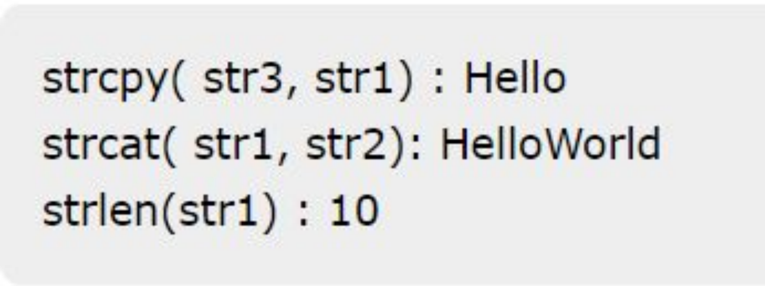
String Functions

Function	Purpose
append()	The append() function adds one string to the end of another.
Compare()	The compare() function returns an integer value indicating the comparison result.
Substr()	Use the substr() function to extract a substring from a string.
Find()	The find() function returns the position of the first occurrence of a substring.
Replace()	Use the replace() function to modify a part of the string.
insert()	The insert() function adds a substring at a specified position.
erase()	Use the erase() function to remove a part of the string.

Example



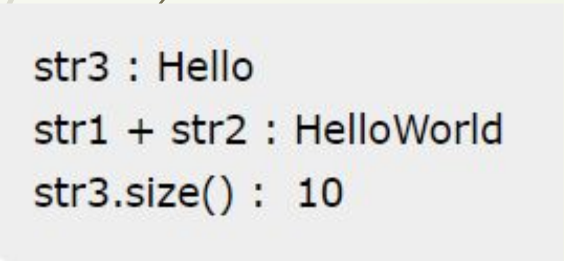
```
#include <iostream>
#include <cstring>

using namespace std;

int main () {

   char str1[10] = "Hello";
   char str2[10] = "World";
   char str3[10];
   int  len ;

   // copy str1 into str3
   strcpy( str3, str1);
   cout << "strcpy( str3, str1) : " << str3 << endl;

   // concatenates str1 and str2
   strcat( str1, str2);
   cout << "strcat( str1, str2): " << str1 << endl;

   // total lenghth of str1 after concatenation
   len = strlen(str1);
   cout << "strlen(str1) : " << len << endl;

   return 0;
}
```



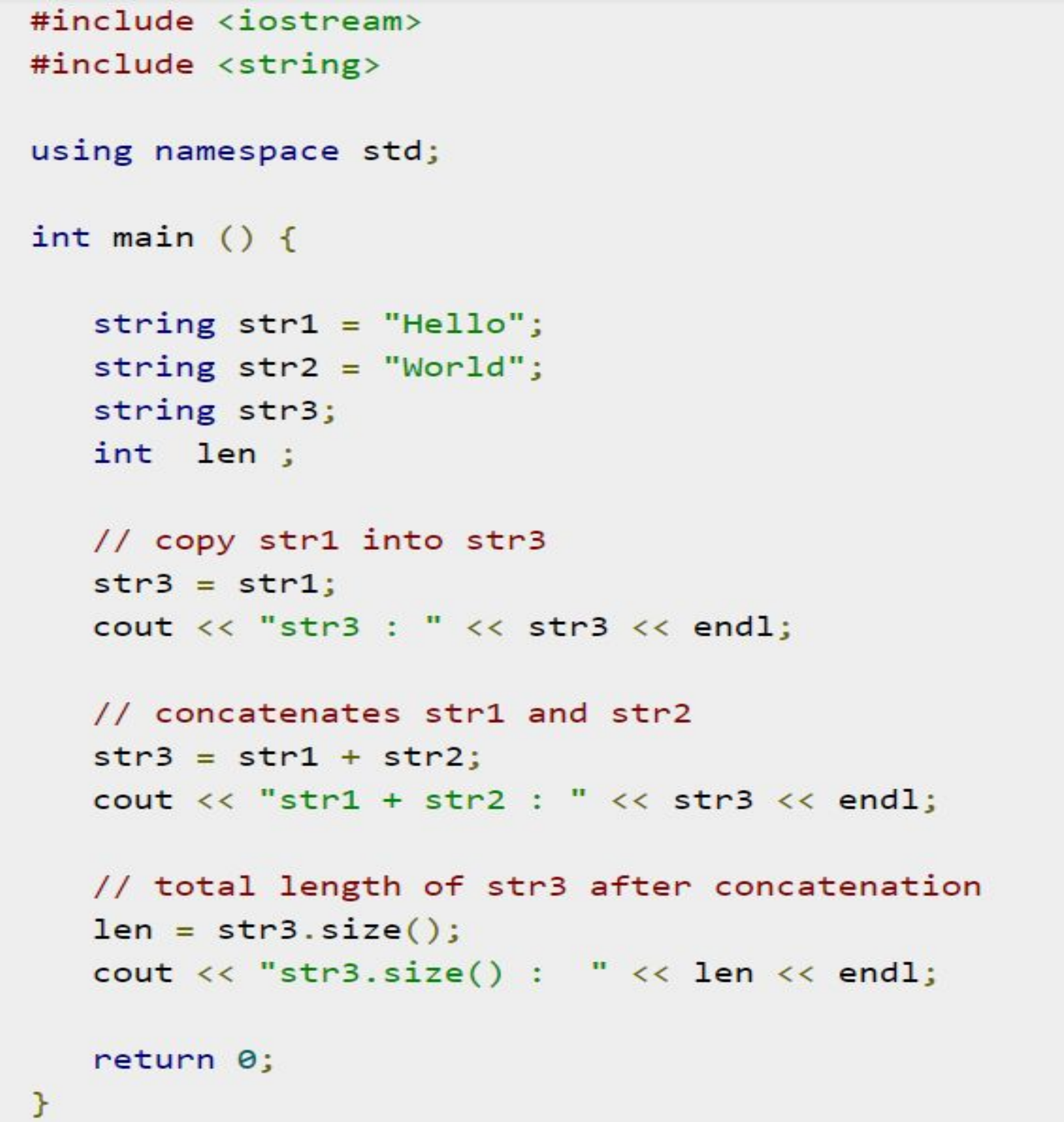
```
strcpy( str3, str1) : Hello
strcat( str1, str2): HelloWorld
strlen(str1) : 10
```

String Class in C++

- The standard C++ library provides a string class type that supports all the operations mentioned before, additionally much more functionality.



```
str3 : Hello
str1 + str2 : HelloWorld
str3.size() :  10
```



```cpp
#include <iostream>
#include <string>

using namespace std;

int main () {

    string str1 = "Hello";
    string str2 = "World";
    string str3;
    int  len ;

    // copy str1 into str3
    str3 = str1;
    cout << "str3 : " << str3 << endl;

    // concatenates str1 and str2
    str3 = str1 + str2;
    cout << "str1 + str2 : " << str3 << endl;

    // total length of str3 after concatenation
    len = str3.size();
    cout << "str3.size() :  " << len << endl;

    return 0;
}
```

Program 2. Program to count total number of vowels and consonants present in a string

```
#include<iostream.h>
#include<stdio.h>
void main( )
{  char string[80]; int count1,count2; count1=count2=0;
  cout<<"Enter a string: ";
  gets(string);
for(int i = 0 ; string[i] != '\0' ; i++)
{ if( string[i] = = 'a' | | string[i] = = 'e' | | string[i] = = ‘i' | | string[i]
= = ‘o' | | string[i] = = ‘u’ | |string[i] = = 'A' | | string[i] = = 'E' | |
string[i] = = ‘I' | | string[i] = = ‘O' | | string[i] == ‘U' )
{count1++;}
else  count2++; }
}
cout<<”The number of vowels is : ”<<count<<endl;
cout<<”The number of consonants is :  ”<<count<<endl;
}
```

```
Enter a string: Grand theatre
The number of vowels is : 4
The number of vowels is :  8
```