

Week 5

Ms. Kanwal Lodhi

string

- String Length

- You can obtain the length of a C++ string using the string class member functions [length\(\)](#) or [size\(\)](#). Both of the methods return the data type [size_t](#) (which is an unsigned integer type), the number of valid characters in the string.

- For example :

```
string s = "Some text";
```

```
cout << "String length is " << s.length() << endl;
```

Ouput : String Length is 9

- String Comparisons

- C++ string objects may be compared using the standard [relational operators](#) ==, !=, >, <, >=, and <=. A C++ string may be compared to either another C++ string or a valid C string, including a string literal. All such relational expressions resolve to the Boolean values true or false.

- Examples:

```
if (s1 > s2)           // Compare two C++ strings
```

```
if ("cat" == s2)      // Compare C string literal and C++ string
```

```
if (s3 != cstr)       // Compare C++ string and array that contains a C string
```

Strings

- **Assignment**

- You can [assign](#) a C++ string object, a C string, or a C string literal to a C++ string object.

- *Examples*

- ```
string s1 = "original string";
```

- ```
string s2 = "new string";
```

- ```
char s3[20] = "another string";
```

- ```
s1 = s2;           // s1 changed to "new string"
```

- ```
s1 = s3; // s1 changed to "another string"
```

- ```
s1 = "yet another string";
```

- ```
// s1 changed to "yet another string"
```

# Strings

- **Concatenation**

- The operator + may be used to concatenate C++ strings. C++ strings, C strings, and string literals may all be concatenated together in any order. The result is a C++ string object that may be assigned to another C++ string object, passed to a function that takes a C++ string object as an argument, printed, etc.

- `string s1 = "Hello";`
- `string s2 = " good ";`
- `char s3[10] = "friend";`
- `s1 = s1 + ", my " + s2 + s3;`  
`// s1 now contains "Hello, my good friend"`

# The Escape sequences

- Special non printing character are used to control printing on the out put device. These are called escape sequence.
- These characters are not printed.
- These are used inside string constants or independently.
- These are written in single or double quote.
- Escape sequence is a combination of backslash '\ ' and a code character.
- The backslash is called the control character.
- Forexample : `cout << "my city is Peshawar\n";`  
Here '\n' transfers the printing control to the next line.
- An escape sequence can be used anywhere in the output stream, means can be used in the beginning, middle or end of string.

# Escape Sequence Characters and its explanation

- **Escape Sequence Characters:**
- **\n:**
  - ‘n’ stands for a new line. It inserts a new line and the cursor moves to the beginning to the next line.
- **\t:**
  - ‘t’ stands for tab. It moves the cursor one horizontal tab from the current tab stop to the next tab stop. For example, an output statement is given below using escape sequence ‘\t’
  - **cout<<"Programming\tDigest";**
  - The output of the statement will be as under:
  - Programming    digest

# Escape Sequence Characters and its explanation

- **'\a':**
  - 'a' stands for alarm. It causes a beep sound in the computer.
- **'\b':**
  - 'b' stands for backspace. It moves the cursor one space back. For example, a statement is given below using '\b' escape sequence.
  - **cout<<"welcome\b";**
  - After execution of this statement, the string "welcome" will be printed and the cursor will move one space back. It means that it will be under the character 'e' of the word "welcome". In the above statement if '\b' is omitted, then the cursor will be at the end of string. If another output statement is used after the above statement, then the printing will be started from character 'e' and 'e' will be deleted.
- **'\r':**
  - 'r' stands for return. It moves the cursor to the beginning of the current line.
- **'\f':**
  - 'f' stands for form feed. It caused the output to feed one paper on the printer attached to the computer.

# Escape Sequence Characters and its explanation

- `\\`:
  - It is used to insert a backslash character in the output. For example, a statement is given below using `'\\'` escape sequence.
  - **`cout<<"\\Programming\\";`**
  - The output will be as under:
  - `\Programming\`
  - If a single backslash character is used, then there will be no effect on the output.
- `'\'`:
  - It is used to insert a single quotation mark in the output. For example, a statement is given below using `'\'` escape sequence.
  - **`cout<<"\'welcom\'";`**
  - The output will be as under:
  - `'welcome'`
- `"\"`:
  - It is used to insert a double quotation mark in the output. For example, a statement is given below using `\"`.
  - **`cout<<"\"welcome\"";`**
  - The output will be as under:
  - `"welcome"`



# The “endl” manipulator

- Manipulators are operators that are used with the put to operator (<<) to control format of data.
- The “endl” is an important and the most commonly used manipulator in C++.
- The “endl” stands for end of line. It has same effect as the escape sequence “\n” . This is a predefined iostream manipulator.
- For example;
  - `cout<< “Pakistan\n”;`
  - `cout<<“Pakistan”<< \n;`
  - `cout<<“Pakistan”<<endl;`These all give the same output.

## Assignment no. 5

- **Example write a program that displays the following output:**
- 'D'      'l'      'g'      'e'      's'      't'
- "Programming"