String Manipulation

This is a brief and a quick lesson about string manipulation. C++ provides very useful features to manipulate strings, this is necessary to understand for making useful programs for real projects.

les_03_code_01.cpp

```
#include<iostream>

using namespace std;

int main()
{
string _str1, _str2, _str3, _str4, _str5;
_str1 = "Pakistan";
_str2 = "The quick brown fox jumps over a lazy dog";
_str3 = "0010-0100-8763-9921";
_str4 = "Hello";
_str5 = "World";

// start at 3 at take all characters
cout<<"_str1.substr(3) "<<_str1.substr(3)<<endl;

// start at 3 and take out 5 characters
cout<<"_str1.substr(3,5) "<<_str1.substr(3,5)<<endl;

// string concatenation
cout<<_str4+_str5;

//compound assignment also work for concatenation
// see for example

_str4 += " ";
_str4 += _str5;

cout<<endl<<_str4;

//resetting _str4
_str4 = "Hello";

//finding location of a value
cout<<endl<<_str4.find('o');

//trying to find location of a value that does not exist
cout<<endl<<_str4.find('f');

//finding location of a value that exist multiple times
cout<<endl<<_str4.find('l'); //returns first location
cout<<endl<<"_str4.rfind('l') = ";
cout<<_str4.rfind('l');
//finding substring
cout<<endl<<_str2.find("fox");

//size of the string
cout<<endl<<_str2.size();

//erase function
string _str6 = "Application and Project Engineer";
_str6.erase(12,4);  //this will erase 4 characters from
                    i.  //location 12
cout<<endl<<_str6;
_str6 = "Application and Project Engineer"; //resetting _str6

_str6.replace(12,3,"for");
cout<<endl<<_str6;


return 0;
}
```

Let's discuss this code in pieces

`substr()` function

`substr()` is used to access a portion of a string (sub-string), this function can be used in the following styles

for a string `_str1`

```
  _str1.substr(3)
```

This will pull all the characters starting from location 3, please note that location numbering starts from 0 and not 1

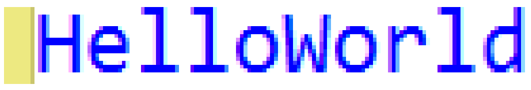
```
_str1.substr(3,5)
```

Start from location 3 and pull 5 characters.

String Concatenation

Two or more strings can be concatenated through concatenation operator `'+'`, see for example

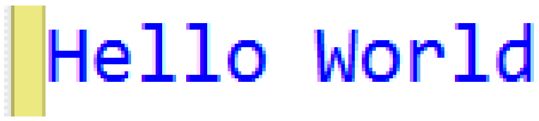
```
cout<<_str4+_str5;
```



```
HelloWorld
```

We can also perform concatenation through compound assignment, for example.

```
_str4 += " ";

_str4 += _str5;

cout<<endl<<_str4;
```

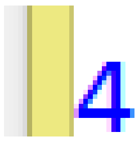


```
Hello World
```

`find()` function

`find()` is used to get the location (starting location for substring) of a character from the string or a substring, for example .

```
cout<<endl<<_str4.find('o');
```



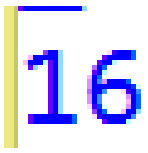
```
4
```

If a character occurs more than once in a string then `find()` will return the location of first occurrence.

`rfind()` (reverse find) can be used to find the location of a character (or a sub string) from the reverse order

`find()` can also be used to find the starting location of a substring, for example

```
cout<<endl<<_str2.find("fox");
```

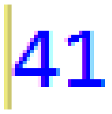


```
16
```

`size()` function

Size of the string can be found using `size()` function, for example

```
cout<<endl<<_str2.size();
```



```
41
```

`erase()` function

A portion of the string can be erased with this function, for example

```
string _str6 = "Application and Project Engineer";
_str6.erase(12,4);
```

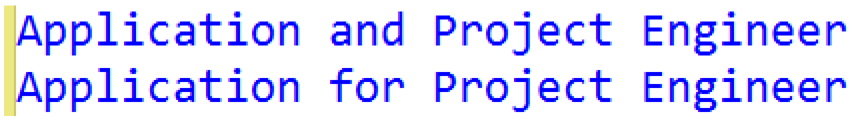
This will erase 4 characters from location 12

`replace()` function

This is used to replace a portion of a string (sub string) with another string, for example

```
cout<<endl<<_str6;
_str6.replace(12,3,"for");
cout<<endl<<_str6;
```

This will replace 3 characters starting from location no 12 with `for`

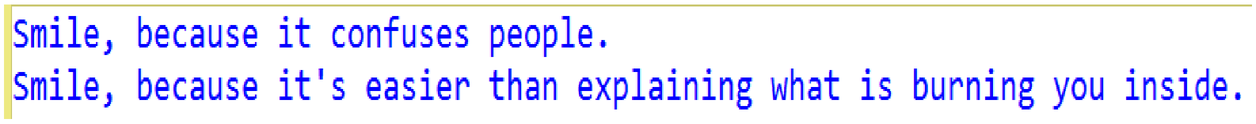


```
Application and Project Engineer
Application for Project Engineer
```

Another example

```
string _str1 = "Smile, because it confuses people.\nSmile, because it's
easier than explaining what is killing you inside.";
string _str2 = "killing";
int loc = _str1.find(_str2);
int width = _str2.size();
_str1.replace(loc,width,"burning");
cout<<_str1;
```

Output

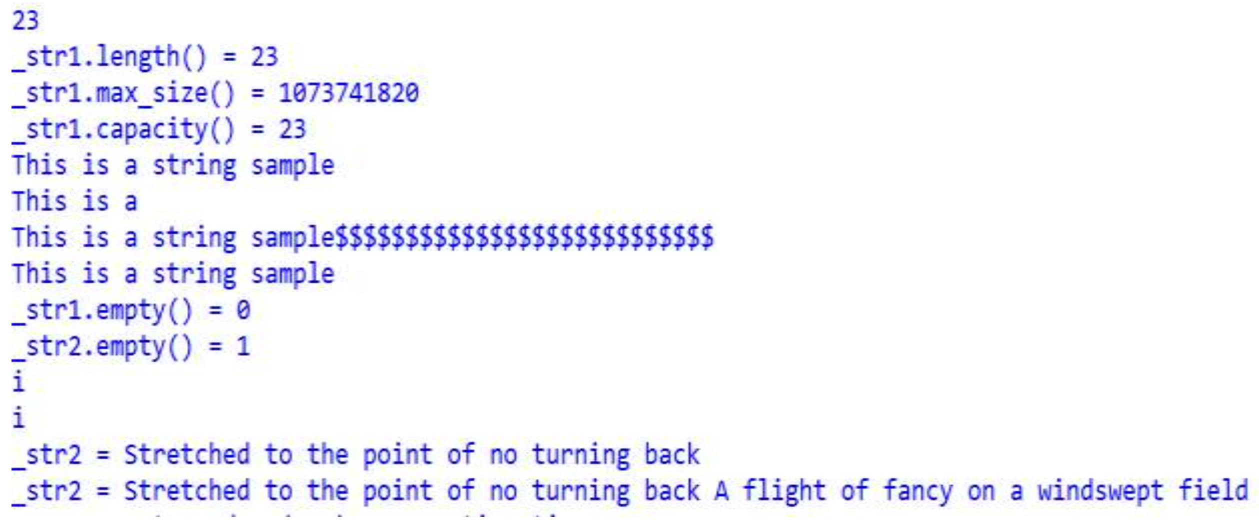


```
Smile, because it confuses people.
Smile, because it's easier than explaining what is burning you inside.
```

les_03_code_02.cpp

```
// les_03_code_02.cpp
// Some more functions and operations for strings
// following functions are explained in this code

//  ----------      length()        ---------- //
//  ----------      max_size()      ---------- //
//  ----------      capacity()      ---------- //
//  ----------      resize()        ---------- //
//  ----------      empty()         ---------- //
//  ----------      at()            ---------- //
//  ----------      append()        ---------- //

// these functions are used as member function
// for more documentations please visit
// http://www.cplusplus.com/reference/string/string/


#include<iostream>

using namespace std;

int main()
{
string _str1, _str2;
_str1 = "This is a string sample";

// To test size and capacity of a string

cout<<_str1.size();
cout<<"\n_str1.length() = "<<_str1.length();
cout<<"\n_str1.max_size() = "<<_str1.max_size(); //finds the maximum possible string size
cout<<"\n_str1.capacity() = "<<_str1.capacity(); //finds the memory consumed

// To change size of the string
cout<<"\n"<<_str1;

// now resizing

_str1.resize(10); //decreasing the size to 10 characters
cout<<"\n"<<_str1;
_str1 = "This is a string sample"; //resetting the actual string

_str1.resize(50, '$'); //resizing to 50 characters with additional character $
cout<<"\n"<<_str1;
_str1.resize(23);   //resizing to the original size
cout<<"\n"<<_str1;

//to test if a string is empty or not

cout<<"\n_str1.empty() = "<<_str1.empty();
cout<<"\n_str2.empty() = "<<_str2.empty();

//Element access
cout<<"\n"<<_str1[5]; //access 5th element of string starting from 0
cout<<"\n"<<_str1.at(5);//access 5th element of string starting from 0

//Modifiers
_str2 = "Stretched to the point of no turning back";
cout<<"\n_str2 = "<<_str2;
//let's modify _str2
_str2.append(" A flight of fancy on a windswept field");
cout<<"\n_str2 = "<<_str2;

// learn the following functions yourself
// ----------   push_back()     ----------
// ----------   pop_back()      ----------
// ----------   assign()        ----------
// ----------   swap()          ----------

return 0;
}
```

Output



```
23
_str1.length() = 23
_str1.max_size() = 1073741820
_str1.capacity() = 23
This is a string sample
This is a
This is a string sample$$$$$$$$$$$$$$$$$$$$$$$$$$$
This is a string sample
_str1.empty() = 0
_str2.empty() = 1
i
i
_str2 = Stretched to the point of no turning back
_str2 = Stretched to the point of no turning back A flight of fancy on a windswept field
```