Section 10 : Strings

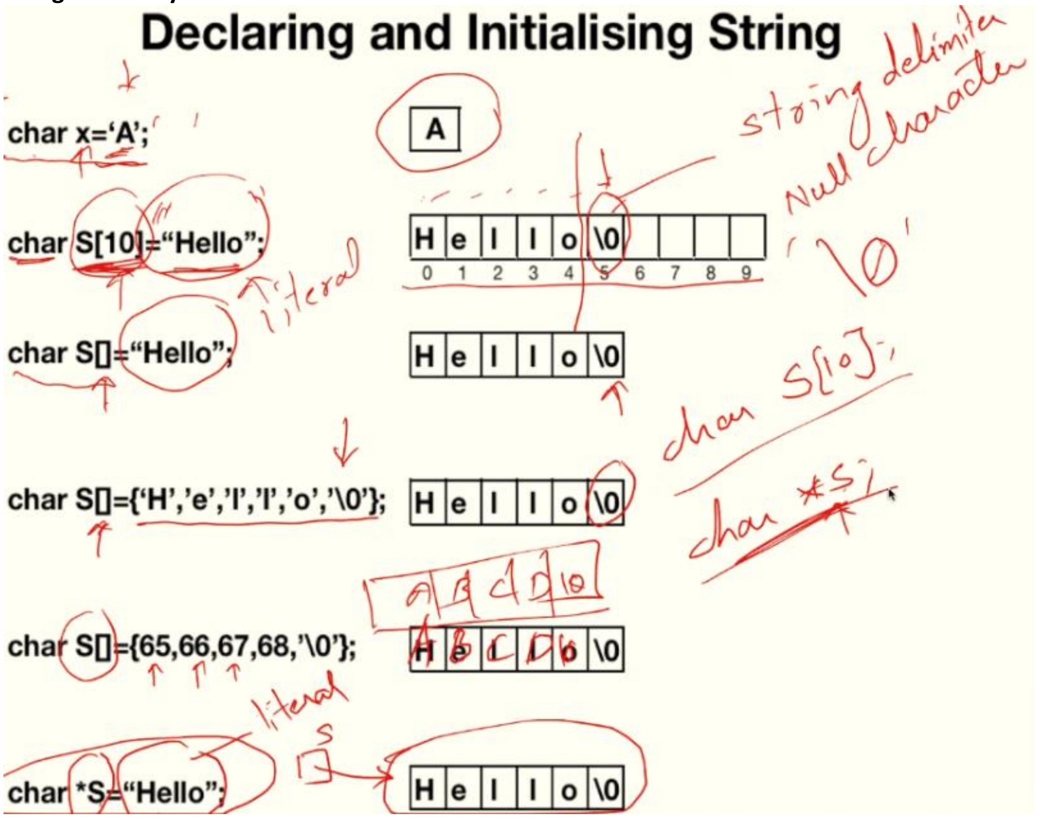
15 April 2022 16:49

© Rajat Kumar
https://www.linkedin.com/in/imRajat/
https://github.com/im-Rajat

Section 10 : Strings

1) Using char Array (available in both c and c++)
2) class string (only in c++)

Using char array :



Declaring and Initialising String

```
char x='A';
char S[10]="Hello";
char S[]="Hello";
char S[]={'H','e','l','l','o','\0'};
char S[]={65,66,67,68,'\0'};
char *S="Hello";
```

- Literals are created in code section
- If we want a string in heap, go for character pointer, if you want in stack, then go for character array.
- Single quotes = char
- Double quotes = string
- Null character = '\0' (or 0 numeric)
- Char *S = "Hello"; // Warning : ISO C++11 does not allow conversion from string literal to 'char*'

char name[20];
cin>>name; // read only first word

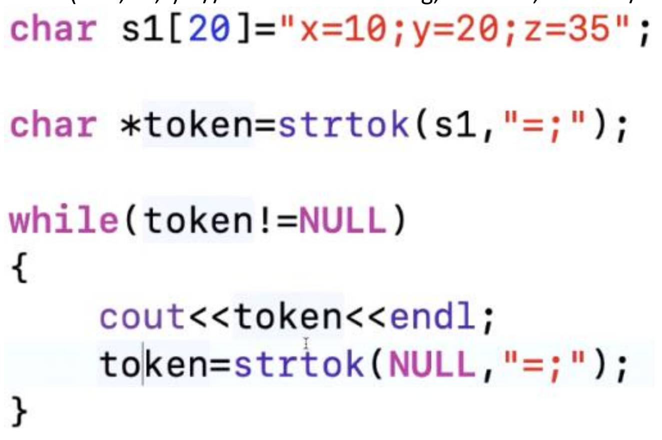
We can use this for sentence:

- Cin.get(name, 20): // get will not read enter key, second string will take that enter as '\n' string
- Use cin,ignore(); after using cin.get(name, 20);
- Cin.getline(name, 20): // use this, use for multiple getlines

Char array/String built-in functions : (#include<ctring> / string.h)

- Strlen(s) // for string length
- Strcat(destination, source) // for concatenate strings, source string will added in destination string, destination will become destination + source.
- Strncat(destination, source, number of letter of second string to concatenate with first)
- Strcpy(destination, source) // copy source string to destination
- strncpy(destination,source, length)
- Strstr(main, sub) // to find substring, will crash if not found. Use if(strstr(s1,s2)!=NULL) {…}
- Strchr(main, char); // find occurrence of a character in string

- Strcmp(str1, str2) ; // compare 2 string, return -ve, 0, +ve
- Strtol(str1, NULL, 10) // string to long int, where 10 (decimal) is base
- Strtof(str1, NULL) // string to float
- Strtok(str1, "=;") // to tokenize a string, where =; is token/delimiter.



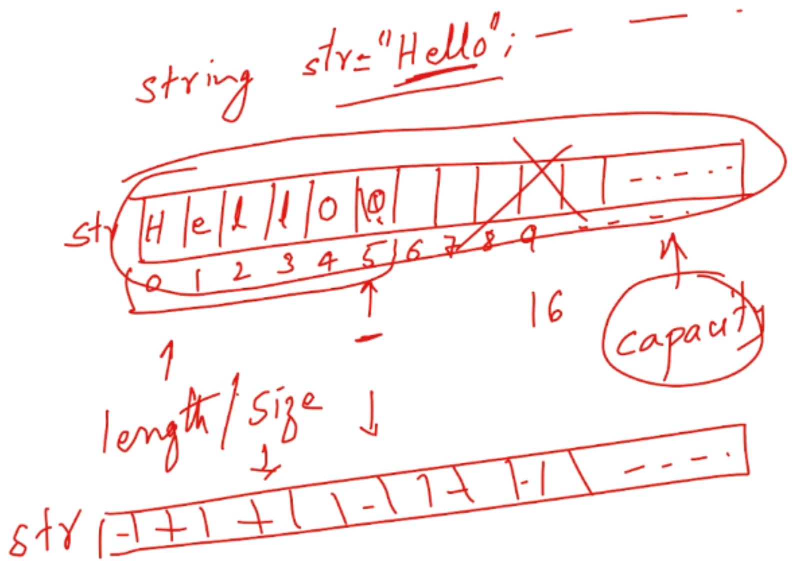
```
char s1[20]="x=10;y=20;z=35";

char *token=strtok(s1,"=;");

while(token!=NULL)
{
    cout<<token<<endl;
    token=strtok(NULL,"=;");
}
```

char *s;
cout<<"Enter a String";
cin>>s;
cin.getline(s, 100); // this will also work

Class String :
#include<string>
string str;
cin>>str; // take only one word
getline(cin,str); // can take sentence

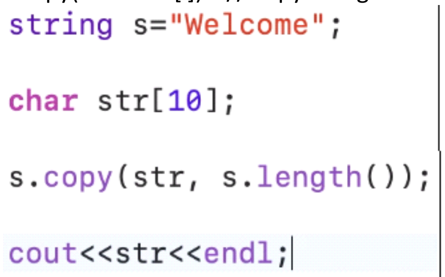


String Class Functions :
s.length( )
s.size()
s.capacity()
s.resize(30)
s.max_size()
s.clear( )
s.empty()

s.append(“Bye")
s.insert(3,"kk") or s.insert(3, "Apple", 2)
s.replace(3,5,"aa") // 3 is starting index, 4 is length from starting index to replace with aa
s.erase() // is same to clear()
s.push_back('z') // insert at the end of string

s.pop_back()   // delete last character of string
s1.swap(s2)   // swap 2 strings

s.copy(char des[ ])   // copy string char array des[]

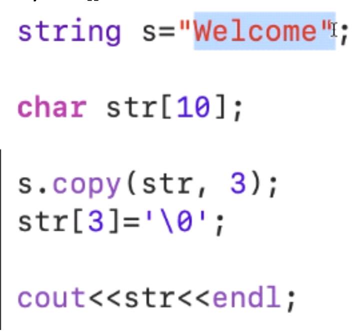


```
string s="Welcome";

char str[10];

s.copy(str, s.length());

cout<<str<<endl;
```



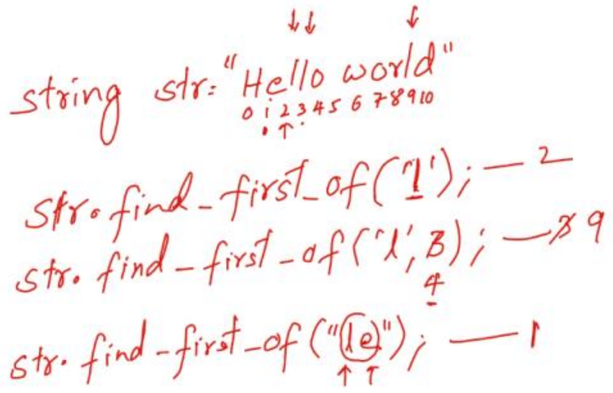
```
string s="Welcome";

char str[10];

s.copy(str, 3);
str[3]='\0';

cout<<str<<endl;
```

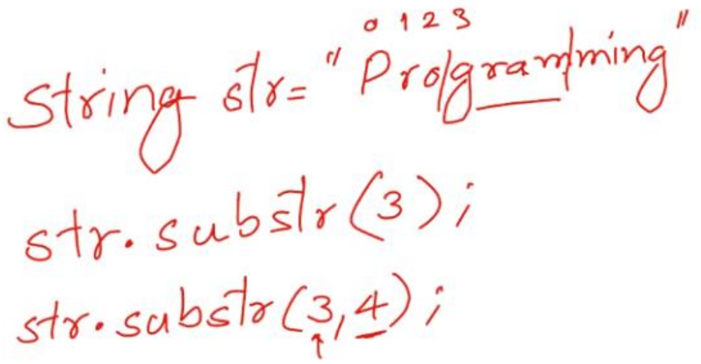
s.find(str) or char   // to find occurrence of string or char and return index
s.rfind(str)   // to find occurrence of string or char from end/right hand side and return index
// if return index is greater than length of string it means it didn't find the string or char
s.find_first_of('a', 3)   // a character to find from last side and start finding from index 3 onwards
s.find_last_of('le')   // search from right hand side, will return index of any of character found first



```
string str="Hello world"
            0 1 2 3 4 5 6 7 8 9 10

str.find_first_of('l'); — 2
str.find_first_of('l', ~~3~~ 4); — ~~3~~ 9
str.find_first_of("le"); — 1
```

s.substr(start,number)   // to extract a portion of string

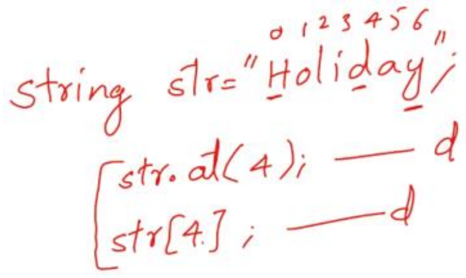


```
             0 1 2 3
string str="Programming"

str.substr(3);
str.substr(3,4);
```

s.compare(str)   // similar to strcmp, compare string in dictionary order and return result as -ve, 0, +ve.

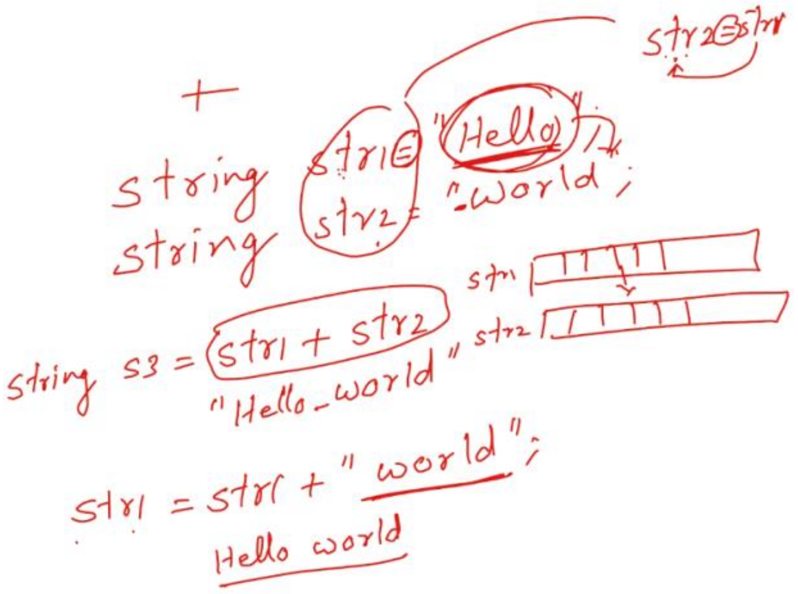
Some operators defined upon string class :
at()



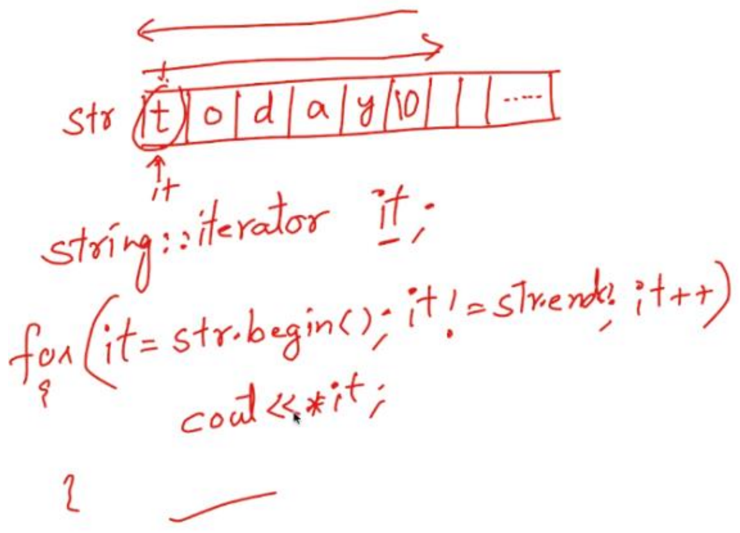
```
            0 1 2 3 4 5 6
string str="Holiday";

str.at(4); — d
str[4]; — d
```

front()   // return first character of string
back()   // return last character of string
[]   // it's overloaded operator
 +
 ll



String Class - Iterators :
string::iterator   // iterator object will work like a pointer to a character in a string (can read and modify)
begin()
end()
reverse_iterator
rbegin()
rend()



Example :
```
string str="today";
string::iterator it;
for(it=str.begin();it!=str.end ();it++)
{
   // cout<<*it;
   *it=*it-32;
}
cout<<str;    // will output TODAY
```

Example :
```
string str="today";
string::reverse_iterator it;
for (it=str.rbegin ();it!=str.rend ();it++)
{
   cout<<*it;
}
// will output yadot
```

Example :

```
string str="today";
for (int i=0;str[i]!='\0';i++)
{
   cout<<str[[i];    // output today
   str[i] = str[i] - 32;
}
Cout<<str;    // output TODAY
```

To create a string of particular length :

```
String str = "";
Str.resize(len);
```