

Strings in C++

The **string** Class

- Definition of Strings
- How to declare strings in C++: the string class
- Operations on strings
 - Concatenation, comparison operators, and []
- Functions of the **string** class
 - **length, size, empty,**
 - **insert, substr, replace, erase, clear, find**
- Useful char functions in the C library <ctype.h>

Definition of Strings

- Generally speaking, a string is a sequence of characters
- Examples: “hello”, “high school”, “H2O”.
- Typical desirable operations on strings are:
 - Concatenation: “high”+“school”=“highschool”
 - Comparisons: “high”<“school” // alphabetical
 - Finding/retrieving/modifying/deleting/inserting substrings in a given string

Strings in C

- In C, a string can be a specially terminated char array or char pointer
 - a char array, such as `char str[]="high";`
 - a char pointer, such as `char *p="high";`
- If a char array, the last element of the array must be equal to `'\0'`, signaling the end
- For example, the above `str[]` is really of length 5:
`str[0]='h' str[1]='i' str[2]='g' str[3]='h' str[4]='\0'`
- The same array could've been declared as:
 - `char str[5] = {'h','i','g','h','\0'};`
- If you write `char str[4] = {'h','i','g','h'};`, then `str` is an array of chars but not a string.
- In `char *p="high";` the system allocates memory of 5 characters long, stores "high" in the first 4, and `'\0'` in the 5th.