STRING

- include <string> for using string class
- Sequence of characters can be represented by an array, but string provides a lot of improved methods to handle and manipulate them
- The string class is very large and include many constructors, member functions and operators
  - Creating string objects
  - Reading string objects from keyboard
  - Displaying string objects on screen
  - Finding a substring from a string
  - Modifying string objects
  - Comparing string objects
  - Adding string objects
  - Accessing characters in string
  - Obtaining the size of string

STD:: STRING VS CHARACTER ARRAY:

- A character array is simply an array of characters that can be terminated by a null character. A string is a class that defines objects that be represented as a stream of characters.
- The size of the character array has to be allocated statically; more memory cannot be allocated at run time if required. Unused allocated memory is wasted in the case of the character array. In the case of strings, memory is allocated dynamically. More memory can be allocated at run time on demand. As no memory is preallocated, no memory is wasted.
- There is a threat of array decay in the case of the character array. As strings are represented as objects, no array decay occurs.
- Character arrays do not offer many inbuilt functions to manipulate strings. String class defines a number of functionalities that allow manifold operations on strings.

INPUT FUNCTIONS

Function	Definition
getline()	This function is used to store a stream of characters as entered by the user in the object memory.
push_back()	This function is used to input a character at the end of the string.
pop_back()	Introduced from C++11(for strings), this function is used to delete the last character from the string.

CAPACITY FUNCTIONS

Function	Definition
capacity()	This function returns the capacity allocated to the string, which can be equal to or more than the size of the string. Additional space is allocated so that when the new characters are added to the string, the operations can be done efficiently.
resize()	This function changes the size of the string, the size can be increased or decreased.
length()	This function finds the length of the string.
shrink_to_fit()	This function decreases the capacity of the string and makes it equal to the minimum capacity of the string. This operation is useful to save additional memory if we are sure that no further addition of characters has to be made.

ITERATOR FUNCTIONS

Function	Definition
begin()	This function returns an iterator to the beginning of the string.
end()	This function returns an iterator to the end of the string.
rbegin()	This function returns a reverse iterator pointing at the end of the string.
rend()	This function returns a reverse iterator pointing at beginning of the string.

MANIPULATING FUNCTIONS

Function	Definition
copy(“char array”, len, pos)	This function copies the substring in the target character array mentioned in its arguments. It takes 3 arguments, target char array, length to be copied, and starting position in the string to start copying.
swap()	This function swaps one string with other.