String functions (#include <string>)

- **getline:** Reads an entire line of characters from input stream and stores it in a string type variable, for example, **getline(cin, myString);**

- **String concatenation + :** At least one of the operands must be a string variable or named constant of string type, the operand can be string type or char type

- **lengh / size function: str.length() or str.size()**
  - returns an unsigned integer value that equals the number of characters in the string
  - *dot notation* is used to call the length function

- **find function: str.find(str)**
  - returns an unsigned integer value that is the beginning position for the first occurrence of a particular substring within the string
  - the substring argument can be a string constant, a string expression or a char value
  - if the substring was not found, function find returns the special value string::npos
- **find function: str.find(str, pos)**
  - When *pos* is specified, the search only includes characters at or after position *pos*, ignoring any possible occurrences that include characters before *pos*.

- **substr function: str.substr(start, length)**
  - returns a substring of a string
  - first argument is an unsigned integer that specifies a starting position
  - second argument is unsigned integer that specifies the length of the desired substring
  - **positions of characters within a string are numbered starting from 0, not from 1**
- **substr function: substr(start)**
  - returns the remaining string starting starting from the "start" location

Example: 1

```
#include <iostream>
#include <string>
using namespace std;

const string STARTING_PHRASE = "In my grandmother's trunk, I packed ";
const string CONNECTOR = " and ";
int main()
{
    string item1, item2, item3, trunk;

    trunk=STARTING_PHRASE;
    item1="my best friend's photo";
    item2="my favorite toy";
    item3="a blue violet";

    trunk = trunk + item1;
    cout << trunk << endl;

    trunk = trunk + CONNECTOR + item2;
    cout << trunk << endl;

    trunk=trunk + CONNECTOR + item3;
    cout << trunk << endl;

    return 0;
}
```

Example 2

```
#include  <iostream>
#include  <string>
using namespace std;

int  main (  )
{
   string  stateName;

   stateName = “Tennessee” ;

   cout  <<  stateName.length( )  << endl;
   cout  <<  stateName.find(“esse”) <<  endl;
   cout << stateName.find(“esse”, 6) << endl;
   cout <<  stateName.substr( 0, 4 ) <<  endl;
   cout << stateName.substr( 4, 2 ) <<  endl;
   cout << stateName.substr( 3, 3 ) <<  endl;
   cout << stateName.substr(5) << endl;

   return 0 ;
}
```

Example 3

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string   name;
    string   lastName, firstName;
    string::size_type    length;
    string::size_type    commaPos;

    name= "John, Smith";
    length = name.length();
    commaPos = name.find(",");

    firstName = name.substr(0, commaPos);
    lastName = name.substr(commaPos+2,
                    length-commaPos-2);

    cout << lastName << ", " << firstName
                    << endl;

    return 0;
}
```

Practice Question:

1. This program (named **upper.cc**) prompts the user to enter a sentence and outputs the sentence with the first letter of each word captalized. For example, the user enters the sentence:
   We drank our coffee the Russian way. That is to say we had vodka before it and vodka afterwards.

   Your program should output:
   We Drank Our Coffee The Russian Way. That Is To Say We Had Vodka Before It And Vodka Afterwards.

2. Find the location of all the appearance of the word “apple” in the line of text read. For example, the user enters the sentence:
   My grandma used all the apples from the our orchard to make delicious apple butter and apple pie.

   Your program should output:
   The word “apple” appeared at locations: 24, 70, 87

3. Prompt the user to enter his/her first name, middle name, and last name in one C++ prompt and input statement. Display the person’s monogram as the first, middle and last initials together as one string.