Classes & Strings

Classes

Technique for implementing new data types and operations on them (data abstraction).

Many useful classes are already defined in class libraries. In particular we use string class.

Similar to built-in data types:
We can declare them

    string myName;

although instead of saying myName is a variable of type string we say
myName is an object of class string.

we can apply assignment to string objects:

    myName = "Dennis Peters";

We can input and output to string objects:

    cout << myName << endl;
    cin >> myName;

In the case of cin, all characters up to the next whitespace character are input into myName.

If you want to input sentences into a string you can use the getline function

ggetline(cin, myName, '\n');

which inputs all chars from cin to myName up to (but not including) the first occurrence of the whatever terminating character you specify.

Member Functions

Objects differ from variables in that they may have member functions

member functions are called using the . operator

len = myName.length();
returns the no. of characters contained in the string.

Some string Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Example</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>myName.length()</td>
<td>Number of characters in string</td>
</tr>
<tr>
<td>at</td>
<td>myName.at(0)</td>
<td>Character at given position (0 is first).</td>
</tr>
<tr>
<td>find</td>
<td>myName.find(&quot;ia&quot;)</td>
<td>Starting position of given string in string.</td>
</tr>
<tr>
<td>insert</td>
<td>myName.insert(6, &quot; K.&quot;)</td>
<td>Insert the new string into the string.</td>
</tr>
</tbody>
</table>

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

int main() {
    string name; // User name, input
    string middle; // middle name, input
    int spacePos = 0; // Position of space in name.
    cout << "Please enter your name, followed by the "Enter" key: 
    getline(cin, name);
    int len = name.length();
    cout << "Length = " << len << endl;
    cout << "Your name is " << name << endl;
    name = "Dennis Peters"; // Change name
    cout << "Updated name is " << name << endl;
    return 0;
}
```
```cpp
cout << "Your name is: \"" << name << "\n";
cout << "Your name contains " << name.length() << " characters."

spacePos = name.find(" ");
cout << "Formal form: " << name.at(0) << ". ";
cout << name.substr(spacePos+1, name.length()) << endl;

cout << "Please enter your middle name: ";
cin >> middle;

name.insert(spacePos+1, middle);
name.insert(spacePos+middle.length()+1, " ");
cout << "Full name: " << name << endl;

return 0;
}

/****************************Nguồn: name.cpp,v $Revision: 1.3 $ $Date: 2001-01-30 20:57:49-03:30 $ $State: Exp $ REVISION HISTORY
* $Log: name.cpp,v $
* Revision 1.3 2001-01-30 20:57:49-03:30 dpeters
* Fixed main header block.
* Revision 1.2 2001-01-30 20:31:03-03:30 dpeters
* Use some other name functions.
* Revision 1.1 2001-01-30 20:22:59-03:30 dpeters
* Initial revision
* *
*ணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணணণ
```