COL 100 Lecture 16

Note Title 9/20/2018

Vector vs. vector

Stanford (SPL)	std:: STL
v. add(value)	v. push_back (value)
v. remove (i)	v. erase ($\underbrace{v.begin() + i}_{iterator}$)
v.insert (i, value)	v.insert (v.begin() + i, value)
v.at(i) v[i]	v. at (i) v[i]

String Streams

#include <sstream>

istringstream
↳ let's you read "tokens" from a string

ostringstream
↳ let's you write tokens to a string buffer

can be a variable

```
istringstream iss ("hello world 123 456");
  string word1, word2, word3;        int x;
   iss >> word1;          // word1 = "hello"
    iss >> word2,          // word2= "world"
if (iss >> x){ }           // x = 123
     iss >> word3          // word3 = "456"
```

~~string ss~~

ostringstream oss;

int age = 42;

int iq = 95;

~~cout~~ oss << "abc's age is ";

1 token (string)

oss << ~~42~~ age;

2nd token (int)

oss << "␣and his iq is ␣";

oss << iq;

string result = oss.str();

cout << result;

iss

istringstream iss(result);

iss >> word1;

oss << "abc's age is ␣" << age;

oss << "␣ and his ip is ␣" << ip ~~<<~~ ~~;~~ << ~~endl~~ "\n";

string s = oss.str();

?

what is the type of oss? ostringstream

```
string l  =  getLine ( " line?" ) ;

    istringstream   iss ( l ) ;

        int   i;
        string   s;

        iss  >>  i
          or
         iss >> s   ?
```

Homework Exercise:

write g SPL functions
like
getInteger
getDoubles

using ~~istringstream~~
stream operator

```
cout<< oss;   X
```

```
string   s =  getLine ("input?");

 istringstream    iss(s);
  string  word;     int   count = 0;
 while  (iss >> word)
  {
        count++;

   }

   cout  <<  count  << endl;
```

≪ _ ≫

getline ($\underbrace{\qquad}_{\text{input stream}}$, $\underbrace{\qquad}_{\text{string variable}}$);

cout << " What is you full name?";
string name;
~~cin >> name ;~~ // X : will not work

get$\underline{\underline{l}}$ine (cin , name);
↳ small

~~input~~

```
istringstream   input( "hello world \n what a nice day \n");
  string  line, line2;
getline (input,   line); // line = "hello world"
  getline (input, line2); // line2 = what a nice day"
  if  (getline (input, line3)) // returns false in this example
  {
     // do something with line 3
   }
```

Exercise: count number of lines in a string

Caution: Do not mix >> and getline()
on the same
input stream!

```
string name; int age;
cout << "How old are you? ";
cin >> age;
cout << "Ad what's you name? ";
~~cin~~ getline (cin, name); ✗ will not work
```

Exercise. write a function string_stats(s) and prints statistics about the data in string s.

```
string s = "Hello I am really enjoying COL100\n
            What a wonderful course this is\n
            I am glad I chose IIT Delhi\n\n
            I am sure this will help me in future\n";
string_stats(s);
       Line 1 :   34 chars , 6 words
       Line 2 :  .32 chars , 5 words
       Line 3 :   32 characters , 8 words.
       Line 4:    0 characters , 0 words
```

Line 5 : 37 chars , 9 words

longest = 37 characters

average = 27.0 characters