C++ and STL Review

- This time we discuss
  - A few C++11 features
  - IO streams
  - String class

- We can only focus on the most important or common features, check a C++/STL book or
- Online materials
  - Course Library->Library->C++ references etc.

New Features in C++11

- Uniform initialization using {}
  - Same syntax for initialization of all data types
  - int I; // I is not initialized
  - int i{}; // I is initialized to zero
  - int *j; // j is not initialized
  - int *j{}; // j is initialized to nullptr
- Initialization of vectors using {}
  - vector<int> vec1 = {10, 20, 30};
  - vector<int> vec2{10, 20, 30};
- How about
  - vector<int> vec3(12); //specifying size of vector
  - vector<int>vec4{12}; // value initialization
  - Curly braces are for value initialization

New Features in C++11

- Keyword *auto*
  - You do not need to specify the type
  - auto i = 20;
  - auto itr = vec1.begin();
  - Compiler must be able to infer
- Range-based for loop

```
int sum = 0;
for (int x : squares) {
        sum += x;
}
```

C++ I/O Streams

- Global stream objects
  - cin: standard input channel, equivalent to C stdin
  - cout: standard output channel, equivalent to C stdout
  - cerr: standard error channel, equivalent to C stderr
- Standard stream operators
  - <<: output operator
  - >>: input operator
- State of streams (iostate)
  - goodbit: everything is OK
  - eofbit: end of file encountered
  - failbit: error; an I/O operation was not successful
  - badbit: fatal error; undefined
  - fail() member function returns true if either failbit or badbit is set

C++ I/O Streams

- Stream state and Boolean conditions
  - operator void*(): check if stream has not run into an error
  - operator !(): check if stream has run into an error

```
if (!cin) {
        …
}


string str;
while (cin >> str) {
        …
}
```

Member Functions

- For input
  - get(): with or with parameter. get next character
  - getline(): get next line in a C string (char *str)
  - A nonmember function getline(cin, string) is more commonly used if you want to the input line to be in a string class
- For output
  - put(): write the character to stream
  - flush(): flush stream buffer
- See example1.cpp and example2.cpp (and example2a.cpp)
  - To compile: make example.x (in general, you can type make program_name.x to compile a program, due to the way we write the makefile)
  - example1.x < testfile

File Access

- Ifstream: input file stream
- ofstream: output file stream

- File streams use the same operators and member functions as I/O streams

```
ifstream file("filename");
ifstream file;
file.open("filename");
```

- See example3.cpp

C++ String Class

- Header file <string>
- Important operators and member functions
  - +=, add character or string
  - +, concatenate two strings
  - []: index operator
  - at(): retrieve element at the specified position
  - clear(): delete all elements
  - empty(): if string is empty
  - substr(): get a substring from the string
  - c_str(): return C character array (C string)
  - size(), length(): return number of characters
- See examples
  - reverse_strings.cpp, split.cpp, split_getline.cpp
  - Example10.pdf, example10.cpp, example10_input.txt,

C++ String Class

- More member functions
  - Operator+=(), append(): append to string
  - Push_back(): add character to string
  - Insert(): insert into string
  - Erase(): delete characters from string
  - Replace(): replace portion of string
  - Find(): find content in string
  - Find_first_of(), find_last_of(), etc: find character in string
- Member constants
  - npos: maximum value for size_t

C++ String Class

- Examples
  - See replacement.cpp
  - Run as replacement.x old_string new_string
  - For a given string, search for old_string and replace it with new_string
    - E.g. replacement.x test TEST
    - Change all test into TEST

String stream Input and Output

- istringstream, derived from istream, reads from a string.
- ostringstream, derived from ostream, writes to a string.
- stringstream, derived from iostream, reads and writes a string.
- You can use the same set of operators and member functions as I/O streams

```
string str;
int num;
istringstream ss("test 25");
ss >> str >> num;
```

- See example4.cpp and example5.cpp
- See conversion.cpp

STL Algorithm copy()

- Function signature

  OutputIterator copy (InputIterator first, InputIterator last, OutputIterator result);

  – Copy elements in the range [first, last) to the range beginning at result iterator

- Note that you need to make sure there is enough space starting from result if it is associated with a container
- Otherwise, you should use some special iterator such as back_insert_iterator or ostream_iterator

- See r2/example6.cpp

STL Algorithm reverse()

- Function signature

  template <class BidirectionalIterator>

  void reverse (BidirectionalIterator first, BidirectionalIterator last);

  – Reverse the order of the elements in the range [first, last)
- Note that this function requires bidirectional iterators
  – Supporting both operator++() and operator—()
- See r2/example7.cpp