STL EXAMPLES

`map`, `list`, `pair`, and `iterator`

Delroy A. Brinkerhoff

STL CONTAINER SUPPORT

ITERATORS

- Sequential container access
- Can access the container's private features, tightly binding the iterator and container
- Overload ++ and -- (forward & backward)
- Container functions return pointers
  - Often used with the arrow operator (->)

THE pair STRUCTURE

- Included with containers
- #include <utility> for standalone

```
template <class T1, class T2>
struct pair
{
    T1 first;
    T2 second;
};
```

WordCount WITH STL CONTAINERS

- ```
  #include <list>
  ```
- ```
  #include <map>
  ```
- ```
  struct w_count
  {
      string    word;
      int       count;
  };
  ```

```
...
int main()
{
    ...
    ifstream    file("alice.txt");
    int         c;
    string      word;

    while ((c = file.get()) != EOF)
    {
        if (isalpha(c))
            word += tolower(c);
        else if (word.length() > 0)
        {
            ...
            word.clear();
        }
    }
    ...
}
```
```

WordCount: list & STRUCT

- `#include <list>`
- `list<w_count> words;`
-
```
list<w_count>::iterator i = words.begin();
while (i != words.end() && word > i->word)
    i++;
if (word == i->word)
    i->count++;
else
{
    w_count node;
    node.word = word;
    node.count = 1;
    words.insert(i, node);
}
```
-
```
for (list<w_count>::iterator i = words.begin(); i != words.end(); i++)
  cout << left << setw(20) << i->word << right << setw(3) << i->count << endl;
```

WordCount: list & pair

- `#include <list>`
- `list<pair<string,int>> words;`
-
```
list<pair<string,int>>::iterator i = words.begin();
while (i != words.end() && word > i->first)
    i++;
if (word == i->first)
    i->second++;
else
    words.insert(i, make_pair(word,1));
```
-
```
for (list<pair<string,int>>::iterator i = words.begin(); i != words.end(); i++)
  cout << left << setw(20) << i->first << right << setw(3) << i->second << endl;
```

WordCount WITH AN STL map AND pair

- `#include <map>`
- `map<string, int>    words;`
- `words[word]++;`
-
```
for (map<string,int>::iterator i = words.begin(); i != words.end(); i++)
  cout << left << setw(20) << i->first << right << setw(3) << i->second << endl;
```