

CSC 215

Math and Computer Science



Iterators

- What are they?
 - Allow a computer programmer to traverse a container(lists, vectors, strings, maps...) of items easily
 - They are basically a pointer
 - Contain the address of some data item
 - Use ++ and -- to traverse through the container

Types of Iterators

- Forward iterators
 - ++ will move your iterator forwards in the container
 - -- will move your iterator backwards in the container
- Reverse iterators
 - -- will move your iterator forwards in the container
 - ++ will move your iterator backwards in the container
- Not all containers have both forward and reverse iterators

I will use the string class for demonstrating iterators.

Declaring an Iterator

- Forward

container::iterator variablename;

Example:

```
string::iterator it;
```

- Reverse

container::reverse_iterator variablename;

Example:

```
string::reverse_iterator rit;
```

Initializing an Iterator

- Member functions of the container class
 - `begin();` returns an iterator to the first character in the string
 - `end();` returns an iterator to the spot just after the last character in the string
 - `rbegin();` returns a reverse iterator to the last character in the string. First char going backwards.
 - `rend();` returns a reverse iterator to the character just before the first character in the string.

Forward Iterator



- String



begin()

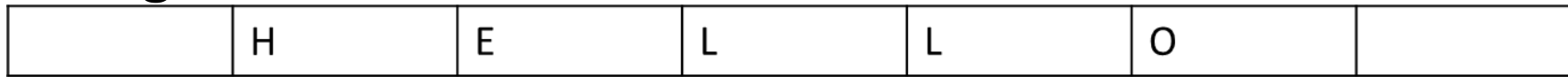


end()

++ moves the iterator 
-- moves the iterator 

Reverse Iterator

- String



rend()



rbegin()

++ moves the iterator ←

-- moves the iterator →

Accessing the Element

- The iterator must be dereferenced to access the item.
- Just like a pointer.
 - `cout << *it;`

Walking Through the List (++)

- Forward iterator

```
string::iterator it;  
string str = "Hello";  
for( it = str.begin(); it != str.end(); it++)  
    cout << *it;  
cout << endl;
```

Outputs: Hello

Walking Through the List (++)

- Reverse iterator

```
string::reverse_iterator rit;  
string str = "Hello";  
for(rit = str.rbegin(); rit != str.rend(); rit++)  
    cout << *rit;  
cout << endl;
```

Outputs: olleH

Walking Through the List (--)

- Forward iterator

```
string::iterator it;  
string str1 = "Hello";  
it = str.end();  
while( it != str.begin() )  
{  
    it--;  
    cout << *it;  
}
```

Outputs: olleH

Walking Through the List (--)

- Reverse iterator

```
string::reverse_iterator rit;  
string str1 = "Hello";  
rit = str.rend();  
while( rit != str.rbegin() )  
{   rit--;  
    cout << *it;  
}
```

Outputs: Hello