

Lamda Functions

```
auto func = [] () {  
    cout << "I'm a Lamda function";  
};
```

Krishna Kumar

Lamda Functions

- Quick and easy way to write functions :)
- Compiler deduces the type, rather than expecting you to specify what kind of return type it is.

// register the class factory function

MyCourseFactory::Instance()->

RegisterFactoryFunction(className,

[](void) -> Course* { return new T();});

Finding an address

```
AddressBook global_address_book;
```

```
vector<string> findAddressesFromOrgs ()  
{  
    return global_address_book.findMatchingAddresses(  
        // we're declaring a lambda here; the [] signals the start  
        [] (const string& addr) { return addr.find( ".org" ) != string::npos; }  
    );  
}
```

Variable Capture with Lambdas

```
// read in the name from a user, which we want to search
string name;
cin>> name;
return global_address_book.findMatchingAddresses(
    // notice that the lambda function uses the the variable
    'name'
    [&] (const string& addr) {
return addr.find( name ) != string::npos; }
);
```

Lamda and STL

```
std::vector<int> v;  
v.push_back( 1 );  
v.push_back( 2 );  
//...  
for ( auto itr = v.begin(), end = v.end(); itr != end; itr++ )  
{  
    std::cout << *itr;  
}
```

Lambda and STL (cont...)

```
std::vector<int> v;  
v.push_back( 1 );  
v.push_back( 2 );  
//...  
for_each( v.begin(), v.end(), [](int val){  
    std::cout << val;} );
```

References

- <http://www.cprogramming.com/c++11/c++11-lambda-closures.html>