# Lab 09

## Resources

- TinyXML2 (http://www.grinninglizard.com/tinyxml2docs/index.html) - Inform
  on TinyXML2, also has the download link.

## Idea

This lab is meant to do your project 3. There will be nothing graded in this lab, so us
time wisely.

## In-lab assignment

There is no assignment, but I will be going over TinyXML2 for those that have been l
trouble with it.

### Steps to use TinyXML2

1. Download the source code found here (https://github.com/leethomason/tinyxm
   github. There is a button that says download zip, this is what you want
2. From that zip file, you are going to extract the tinyxml2.cpp and tinyxml2.h files
   the same directory your other files for project 3 are located. You can then treat t
   files as if you wrote them (Include the .h and add in a target for the make file, et
3. The next problem becomes: How do you use tinyxml?
4. First off, everything in the tinyxml2 files belongs to the *namespace* tinyxml2. So
   will either have to put tinyxml2:: in front of most of the things that you write or
   can put using namespace tinyxml2 at the top of your files.
5. The main classes you need to know about are XMLNode, XMLDocument,

XMLElement, and XMLText.

6. XMLNode is the parent class of every one of these classes. This means everything XMLNode in the end.
7. XMLDocument is going to be your starting point. It has the parsing functions yo need for the xml files you will use:

```
XMLDocument doc;
doc.LoadFile("whatever.xml");
```

These lines will parse the whatever.xml file and build a tree representation of tha for you. How do you access that tree?

8. Remember, everything is a XMLNode. The XMLNode that you might be most insterested in is the *root*. You can get it like so:

```
XMLNode * root = doc.RootElement()
```

Once you have the root, you might want to know what the root is (is it a recipe, a ingredientlist, an ingredient, etc). You can check with the

```
root->Value()
```

This function returns a string representing the root node's value.

9. Once you have the root, the question becomes: How do you get the rest of the tre
10. To go down a level in the tree, you use the FirstChild() function, which returns a *:

```
XMLNode * child = root->FirstChild()
```

Again you can use the Value() function to get this node's value

11. You might want to know when you get to the end of the tree. To do this there is a boolean NoChildren() function which returns true if the node has no children. Alternatively, you could check to see if FirstChild() returns NULL
12. To get to a node's other children, you cannot use some sort of SecondChild() fun Instead, we have functions called NextSibling():

```
XMLNode * sib = child->NextSibling()
```

13. One final thing you may want to know is how can you differentiate between a tag
actual text in an xml file? Tags will be put into the tree as *Elements*. Text will be
into the tree as *Text*. To check if a Node is an *Element* or *Text*, we use the ToEle
or ToText functions:

```
if(node->ToText() != NULL){
  // The node is text
}
else if(node->ToElement != NULL){
  // The node is an element/tag
}
else{
  // The node is neither
}
```