Go over Quiz #7.

Revisit some concepts from worksheet #6, and think about where and how the newer terms you have learned, such as friends, operator overload, templates, and exceptions, fit into this network of concepts?
**Exceptions:**
What is an exception, and how does it differ from the error handling you have been using?

Why would you create your own exception beyond those provided for you?

How does the order of the exceptions caught matter?

**Templates:**
Given the following outline of code, use the vector class from the STL to add a new integer to a vector of integers.

```cpp
#include<iostream>
#include<vector>

using namespace std;

int main(){

}
```

*If you have time...*
How would you create a templated class called `array` and provide the same functionality as `push_back()` in the vector templated class from the STL?

```cpp
template<class element>
class array {
    private:
        element *a;
        int size;
    public:
        void add(const element &item);
    };

template<class element>
array<element>::add(const element &item){

}
```

What about if you want to provide functionality for adding to the front in this class?