**CODE IS ALSO IN MY .CPP FILE**

**Question 1**

vector<int> dataItems(10);

dataItems.at(100);

Unhandled exception at 0x76C43FC8 in Lab 12 Exceptions.exe: Microsoft C++ exception: std::out\_of\_range at memory location 0x00EFFC54.

**Question 2**

try

{

dataItems.at(100);

}

catch (out\_of\_range e)

{

cout << e.what();

}

Prints out invalid vector<t> subscript

**Question 3**

try

{

dataItems.at(100);

}

catch (exception& e)

{

cout << e.what();

}

Prints out invalid vector<t> subscript

**Question 4**

int StringToInteger(const string& input) {

stringstream converter(input); // feed string into a stringstream for re-reading

int result;

converter >> result; // attempt to read as an int

// converter.fail() will be true here if the string isn't a number

char leftover; // See if anything's left over.

converter >> leftover;

// if converter.fail() is NOT true here, it means that there was more than just a number in the string

return result;

}

**Question 5**

invalid\_argument ex("Cannot parse " + input + " as an integer.");

try

{

if (converter.fail())

{

converter >> result; // attempt to read as an int

throw ex;

}

}

catch (invalid\_argument ex)

{

cout << ex.what() << endl;

}

char leftover; // See if anything's left over.

//converter >> leftover;

// if converter.fail() is NOT true here, it means that there was more than just a number in the string

try

{

if (!converter.fail())

{

converter >> leftover;

throw ex;

}

}

catch (invalid\_argument ex)

{

cout << ex.what() << endl;

}

**Question 6**

try

{

StringToInteger(hello);

}

catch (exception ex)

{

throw ex;

}

1. **It’s better to throw the exception back in main in this case because if there is a problem with the returned value the throw statement in main can catch that problem.**

**Question 7**

void what(string);

void what(string message) {

cout << message << endl;

}

**Question 8**

try

{

if (converter.fail())

{

converter >> result; // attempt to read as an int

throw ex;

}

}

catch (invalid\_argument ex)

{

string m = "The string isn't a number.";

//cout << ex.what() << endl;

what(m);

}

char leftover; // See if anything's left over.

//converter >> leftovewr;

// if converter.fail() is NOT true here, it means that there was more than just a number in the string

try

{

if (!converter.fail())

{

converter >> leftover;

throw ex;

}

}

catch (invalid\_argument ex)

{

string m = "There was more than just a number in the string.";

//cout << ex.what() << endl;

what(m);

}

**Question 9**

try

{

StringToInteger(hello);

}

catch (exception ex)

{

string m = "Exception thrown in main!";

throw ex;

what(m);

}