Evan Akers

Smit Patel

Kyle O’Connor

Saylee Dharne

Lab Report 5: Exceptions and Templates

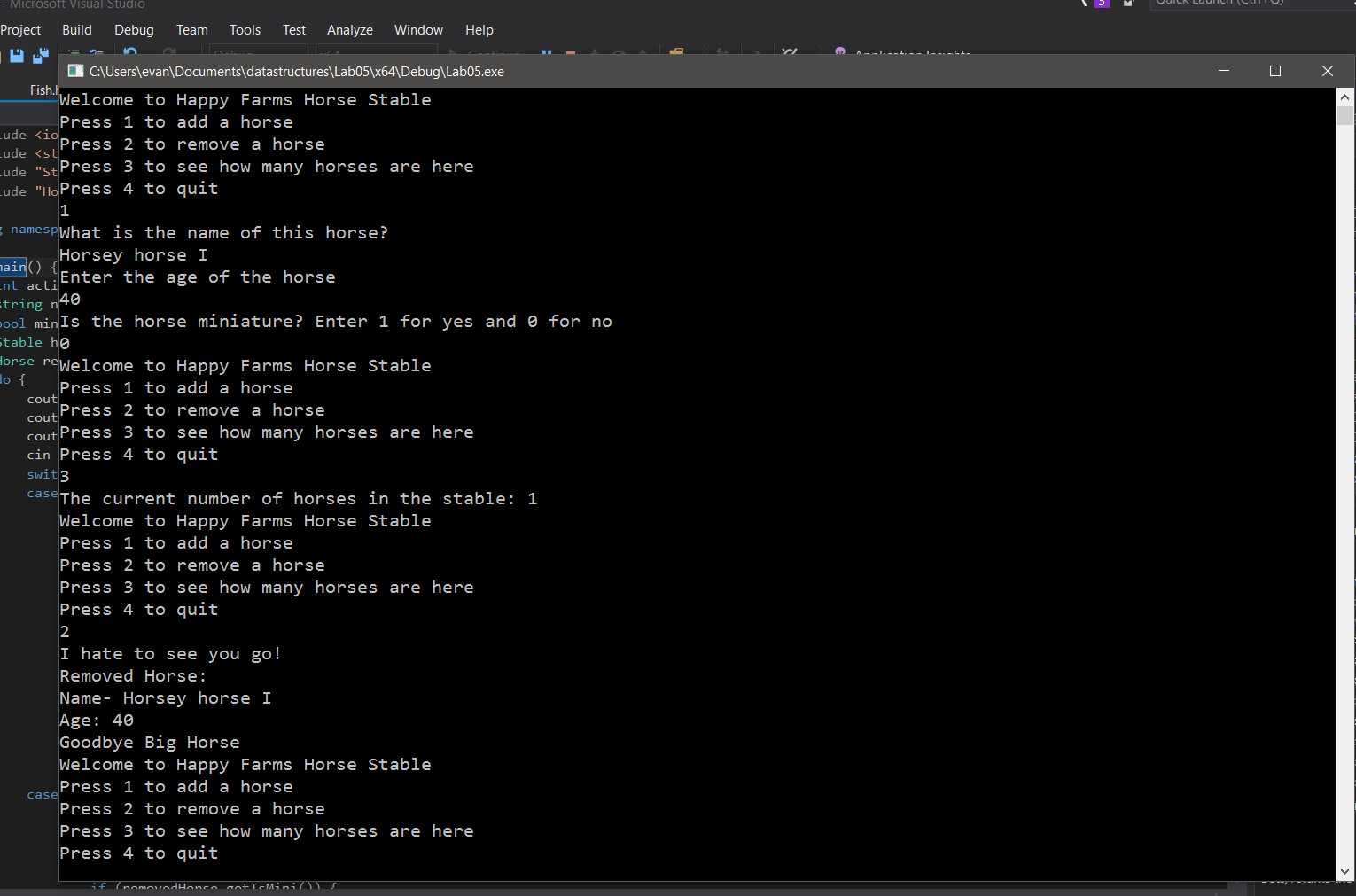
Exceptions are vital to being able to write scalable and maintainable code for any one in a relevant field. Being able to catch and throw exceptions allows code not to crash every time an unexpected input or event occurs. Catching errors allows the code to deal with unexpected events such as full arrays or bad user inputs. Templates are very useful for saving programmer time. Templates allow a class or function to use different data types, from primitives to classes you made. You can build data types, such as containers like linked lists that do not need a datatype defined and they can be used for many different data types.

Task 1

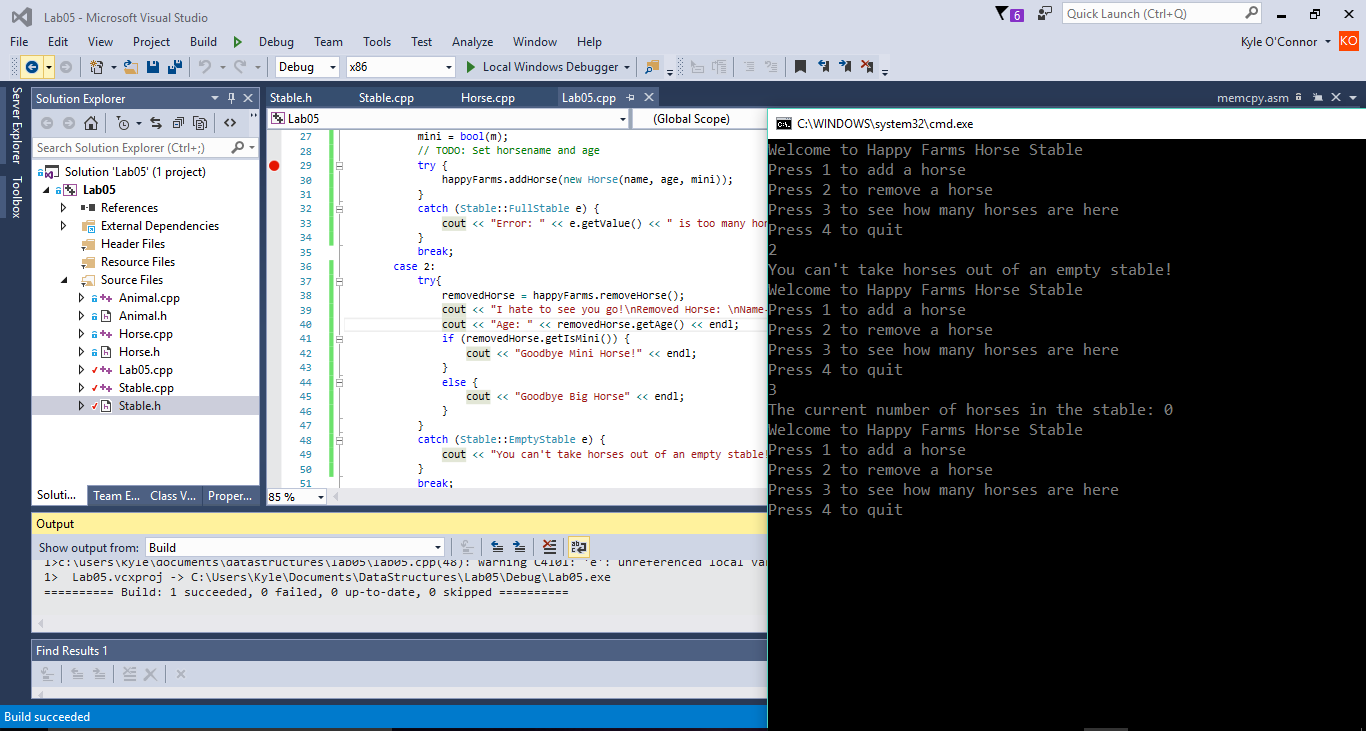
We designed the add() function of the Stable class to take in a Horse and add a Horse into the array of horses in the stable at the next empty spot in the table. We increment currentHorseNum as it represents the next open spot available in the stable.

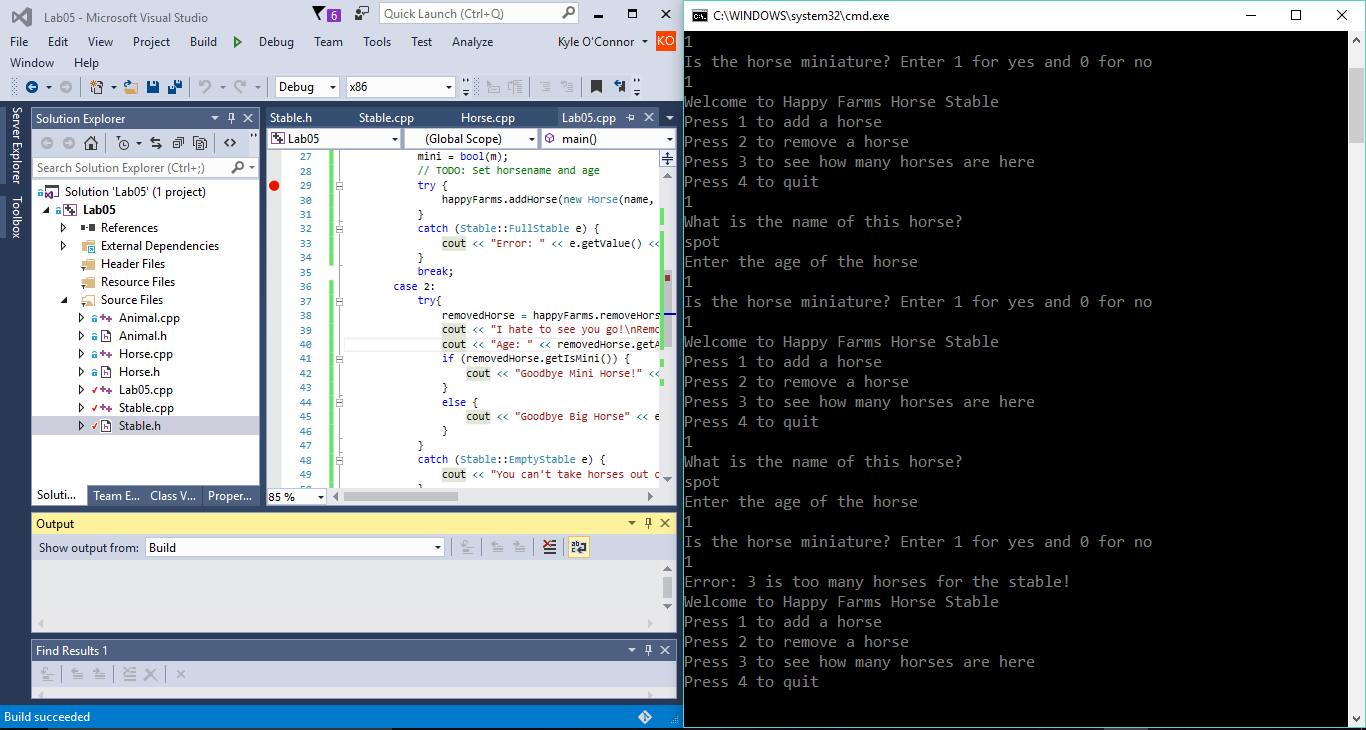
We designed the remove() function to decrement currentHorseNum to access the Horse in the last non-empty position of the stable and return that Horse. We decrement currentHorseNum as it represents the next open spot available in the stable. After decrementing currentHorseNum, we assume that the value of the previous Horse has been cleared, because the next time that the Horse in that position is accessed will be to overwrite that value of Horse.

Task 2



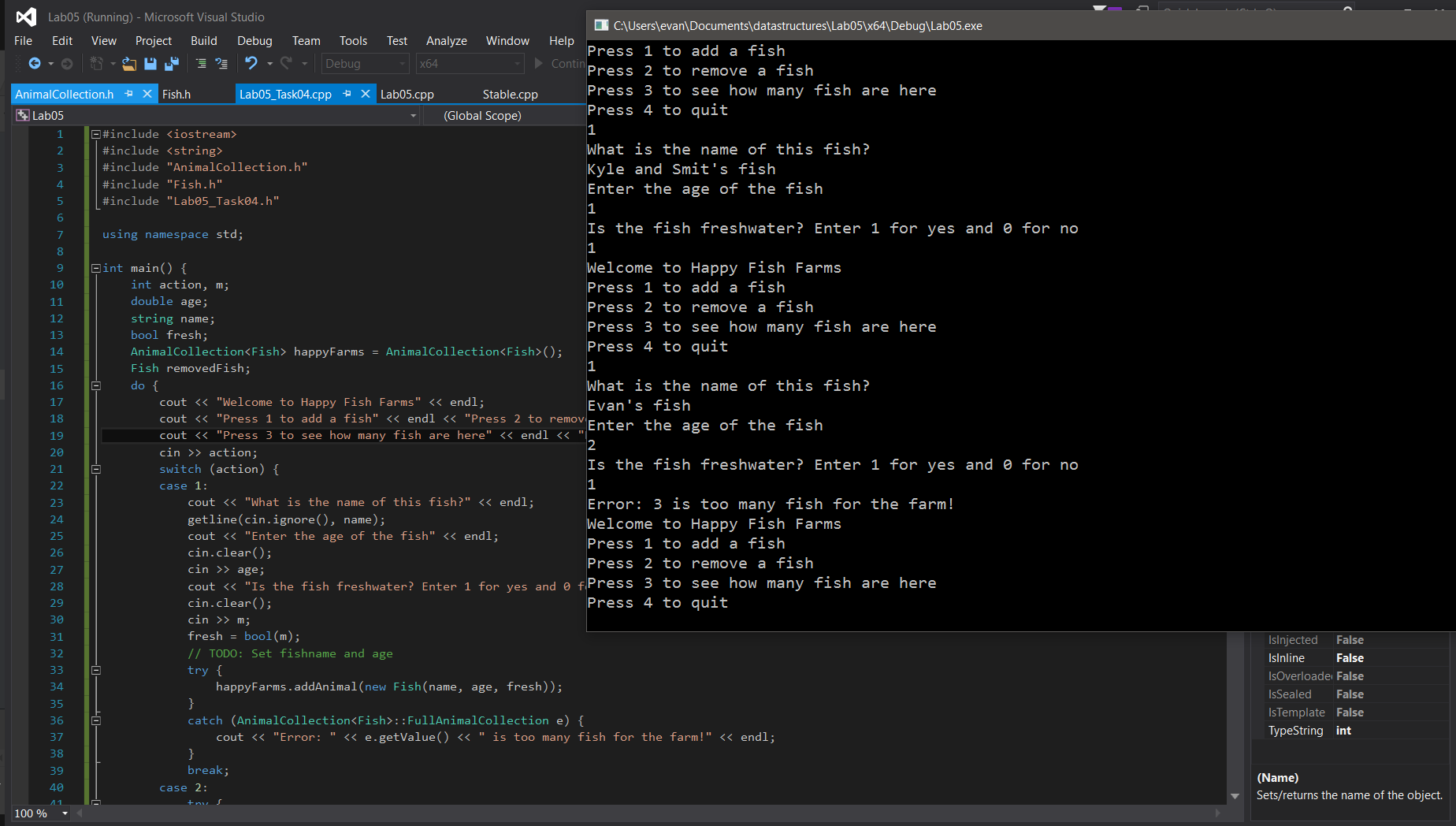
Task 3

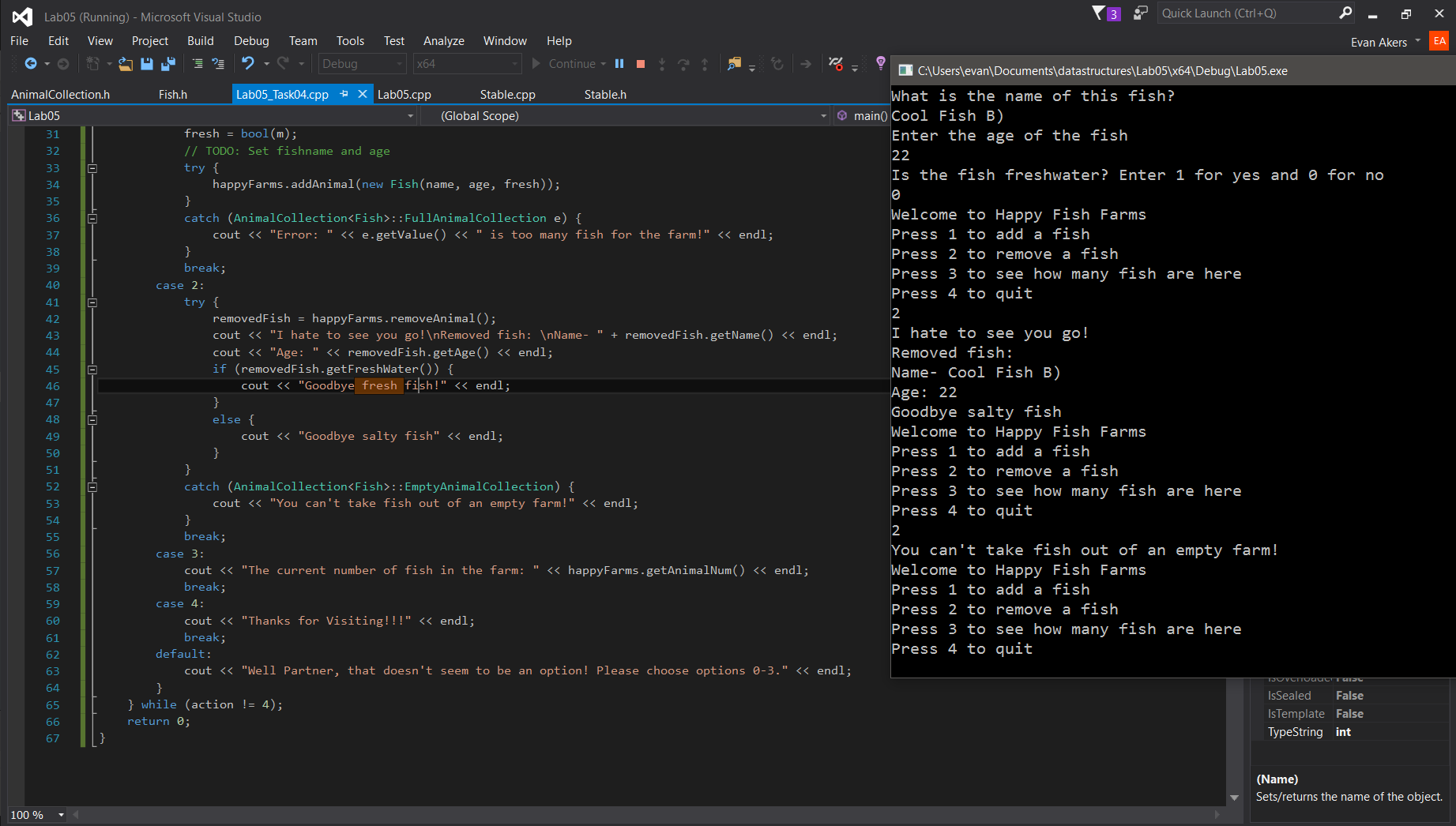


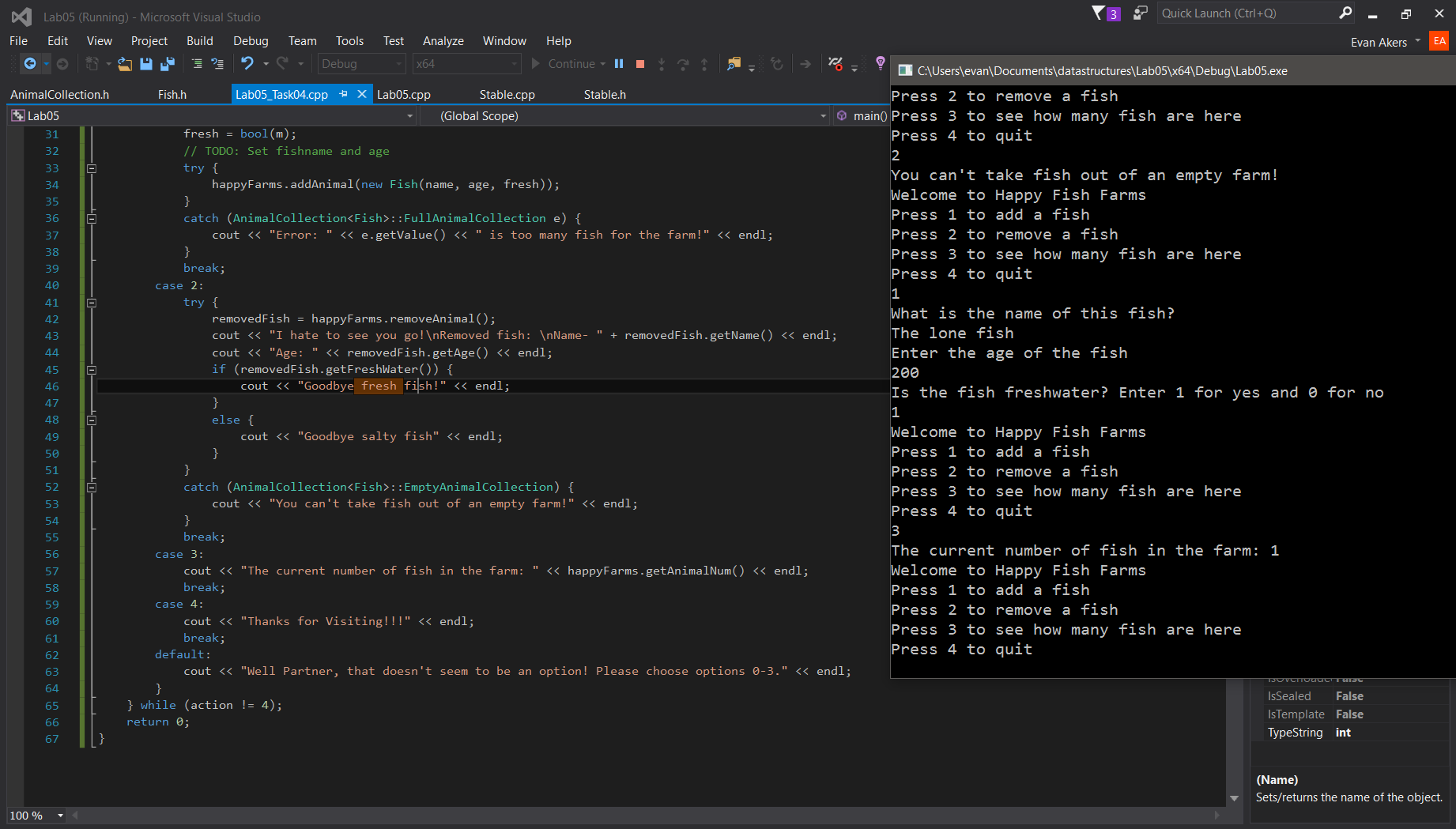


Traditional error handling requires the program using the object and its functions to tie in closely to prevent crashes. For example, in the animal farm case. You would have to keep track of how many animals are in to prevent a crash. If the member functions can throw exceptions, the main program can catch them, preventing the need of the main method from keeping track of attributes that the class should do.

Task 4







The advantages of using templates is having code that is adaptable to different object types. The animal collection can hold any object type, with only one class declaration and definition. I could use the animal collection for any animal class I create, or any data type (assuming I change the main method for the member functions of that specific object).

Compilation Instructions

This has been tested by creating a new project within Visual Studios with the following options:

Win32 Console Application

Create directory for solution OFF

Empty project ON

Precompiled header OFF

SDL OFF

Then:

Add the following files to the projext:

1. Lab05.cpp
2. Horse.cpp to the project
3. Horse.h to the project
4. Stable.cpp to the project
5. Stable.h to the project

Build and run

For task 4:

1. Lab05\_Task04.cpp
2. Fish.cpp to the project
3. Fish.h to the project
4. AnimalCollection.cpp to the project

Build and run.

Contribution of Team Members

Each person wrote the portion of the lab report for their task,

Kyle O’Connor did task 1

Saylee Dharne did task 2

Smit Patel and Kyle O’Connor task 3

Evan Akers did task 4