**Data Structures Lab Report 5**

Group members: Vivek Kunapareddy, Yuan Cheng

**Objectives:**

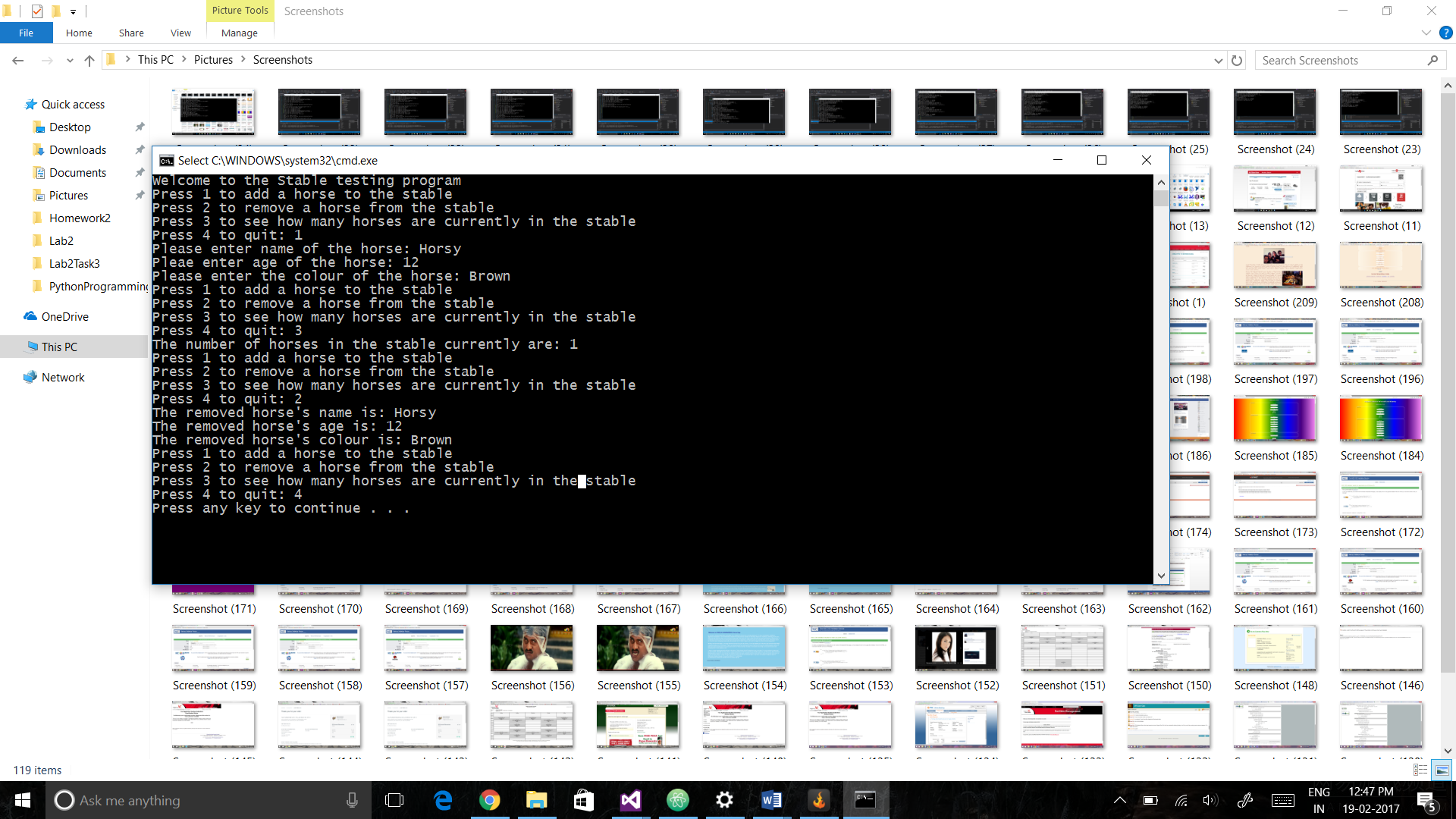
The objectives explored in this lab were C++ templating and exception handling. These concepts will be important when transitioning into the workforce as templates allow programs to be extensible in nature and allow changes to the program later on. Exception handling is also important in software engineering as it allows programs to continue execution and handle errors gracefully.

**Task 1:**

The add and remove methods for the Stable class were defined in terms of an array. Since the class itself has an array of elements, we just moved the pointer to the next empty place in the array instead of deleting elements in the array. Doing so reduces the amount of work needed. For the add method as well, the pointer is used and then incremented, hence overwriting the other elements.

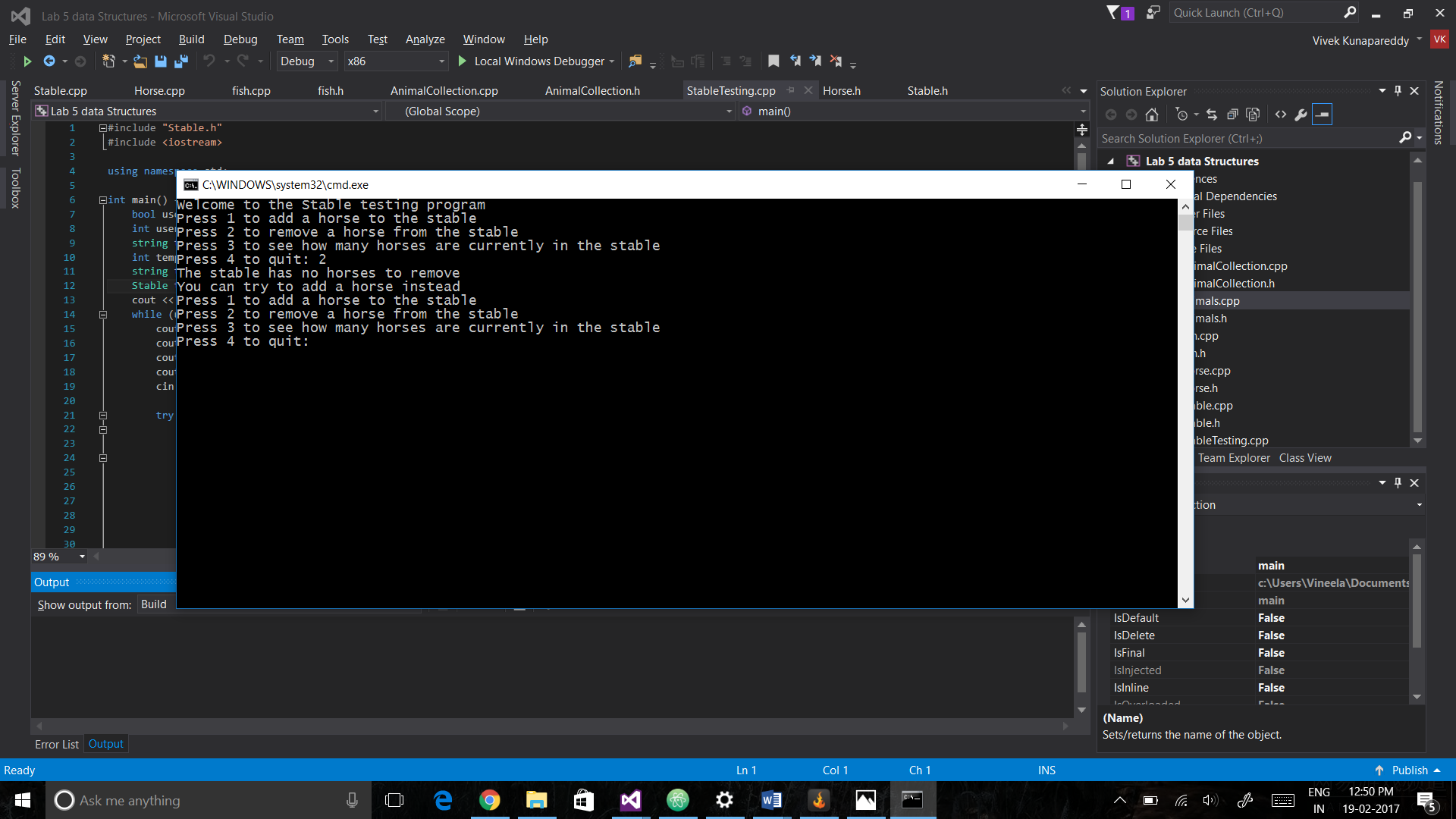
**Task 2**

Screenshot of all functions:

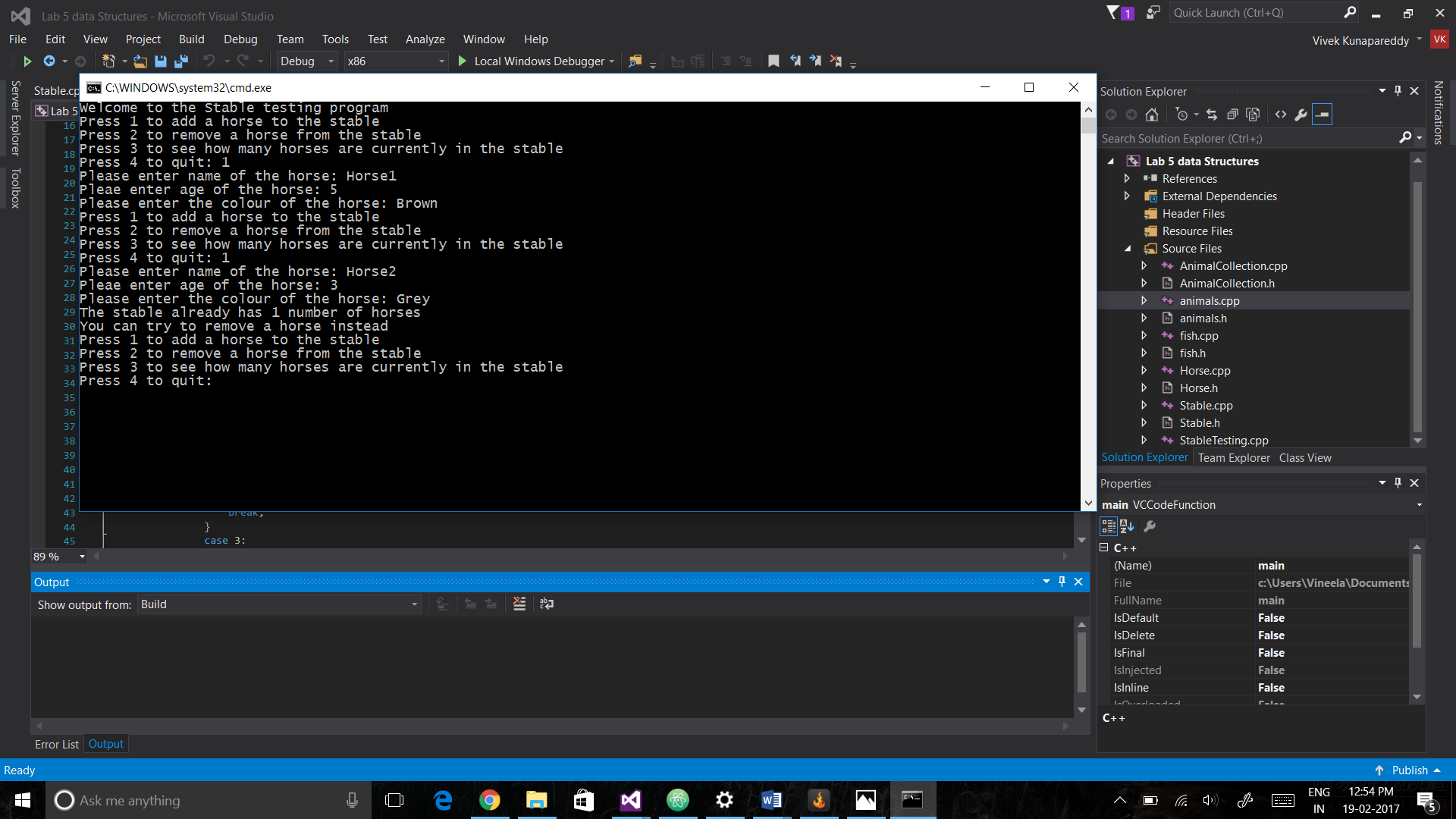


**Task 3:**

Empty Stable:



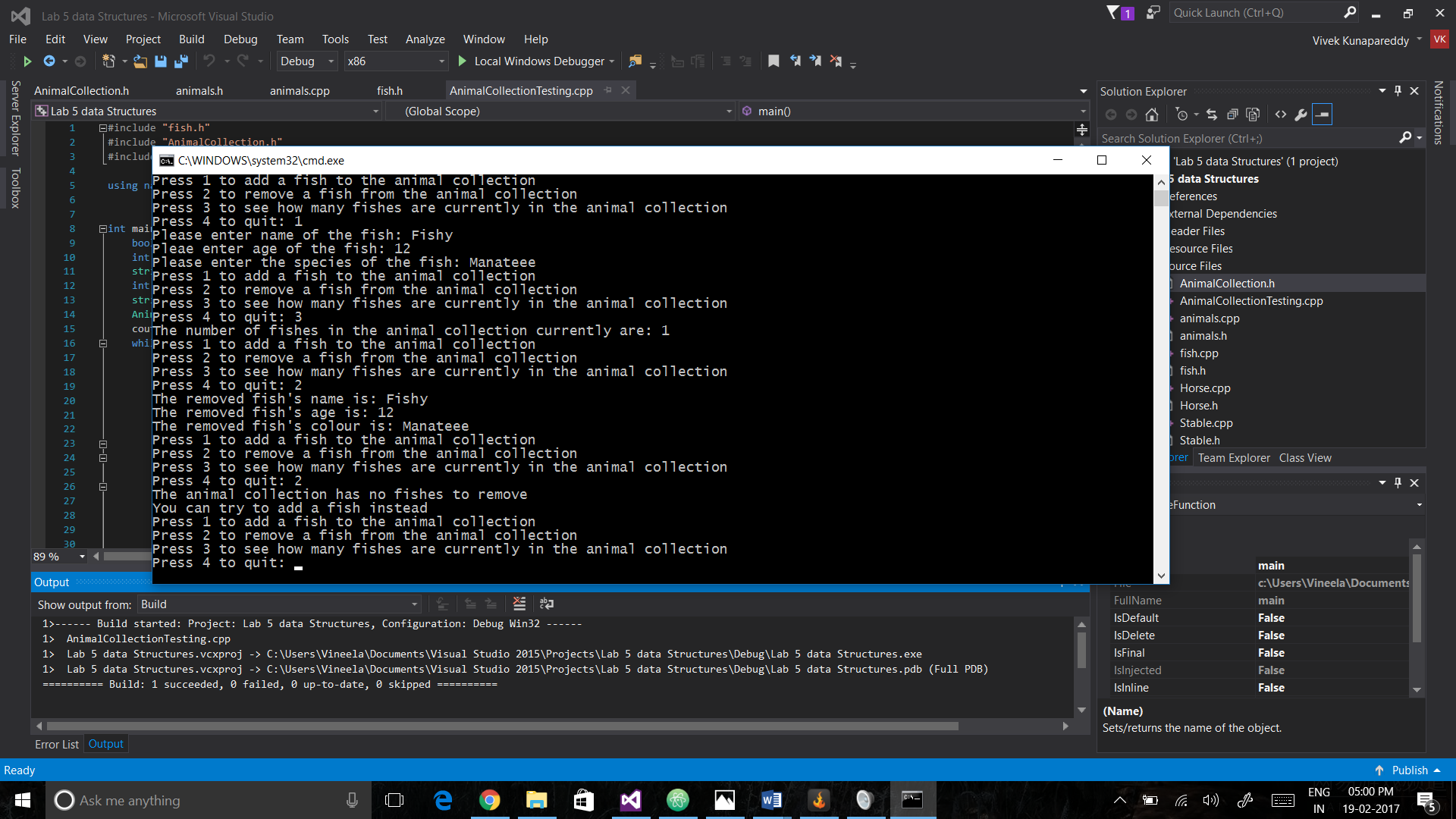
Full Stable:

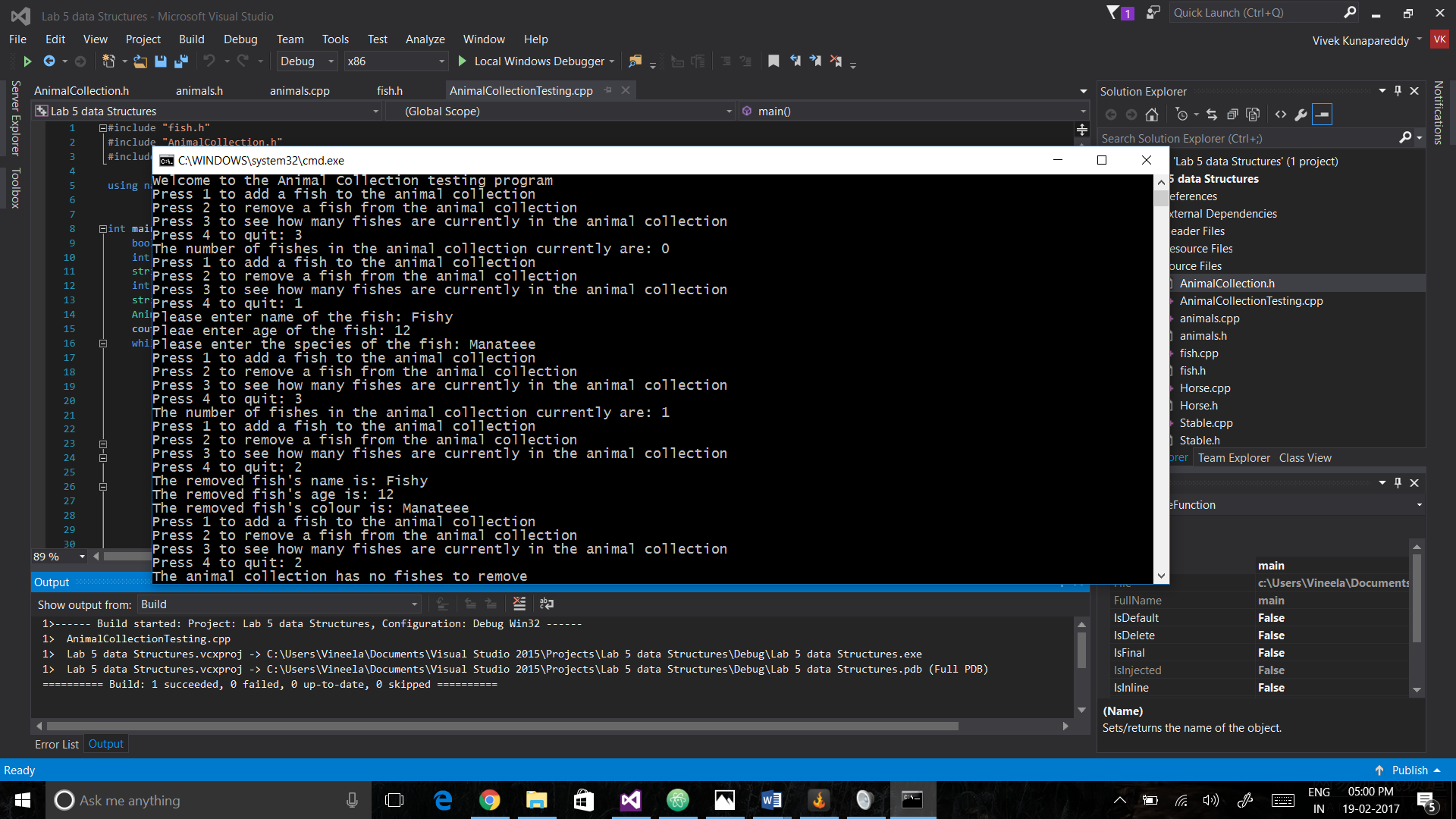


**Advantages of exception handling:**

The exception handling method of errors allows us to have a greater handle on how error handling is treated. It allows execution of the program to continue as compared to traditional error handling which forces program abortion. Also, it allows us to tailor our responses based on the type of error occurred.

**Task 4:**





**Advantages of templates:**

The advantages of templates are very clear. When creating containers or classes which simply need another class, we can create them to use any class we declare in the future. Thus, we use the very basic principle of OOP: polymorphism by using them. Finally, we can use the same class to treat different data types differently by using class-specific templates

**Group contributions:**

All the programming and the debugging were done together