**Data Structures Lab Report 4**

Group Members: Vivek Kunapareddy, Yuan Cheng

**Task 1:**

Animals.h was declared with the member variables as private and all other functions were declared as public, with a single virtual function called move.

In our expectations all the member functions defined as public will be available to any derived class with the base as animals.

The virtual function move() should be available to derived classes only when overridden.

The function eat() defined in the animals class will be available unless overridden in the derived class.

**Task 2:**

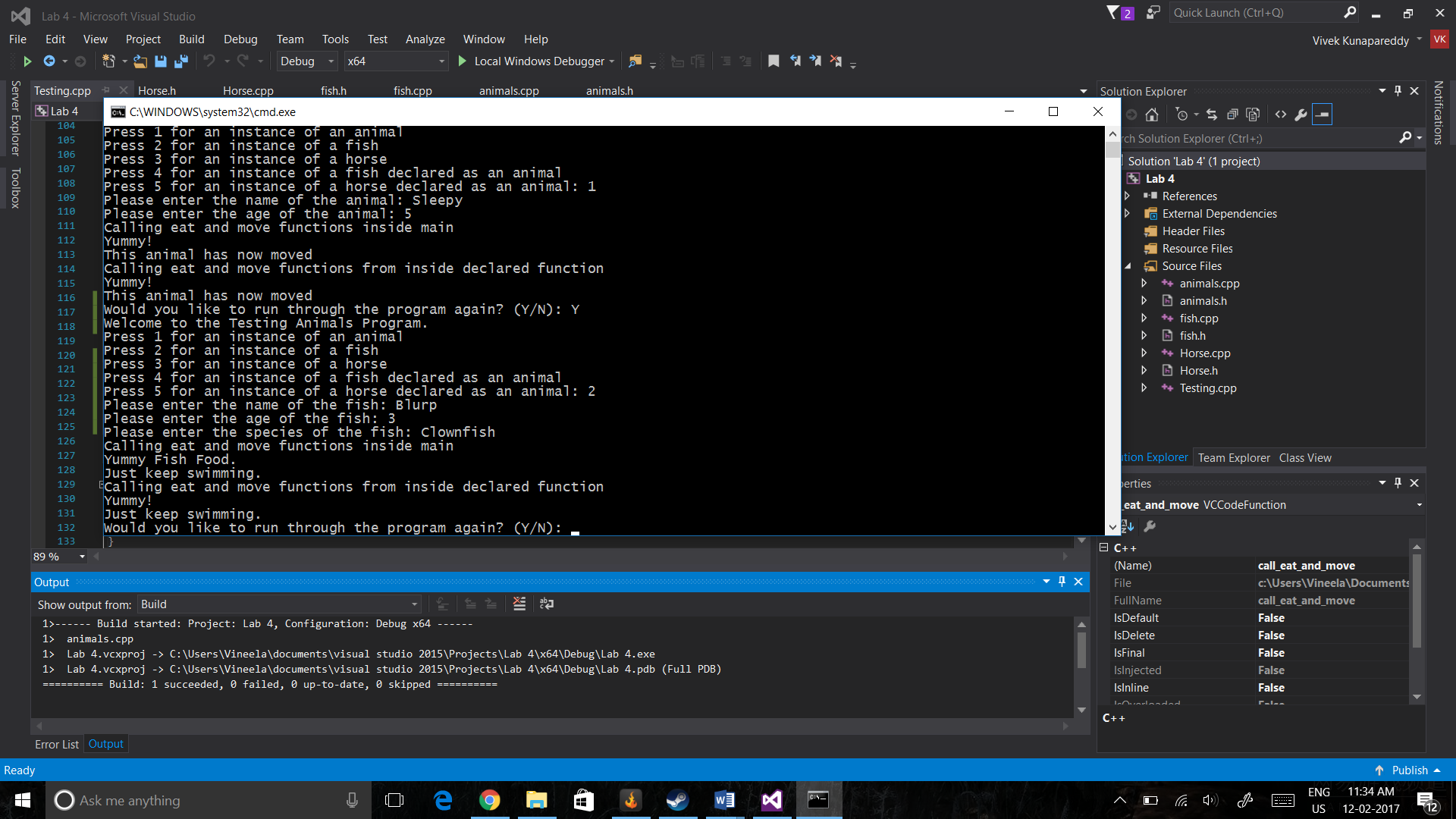
The availability of member variables:

Fish and Horse derived from Animal: They will have access to all the public member functions including move() and eat(). They will not have access directly to the private members name and age. They can override move() and eat() as well in their own declarations.

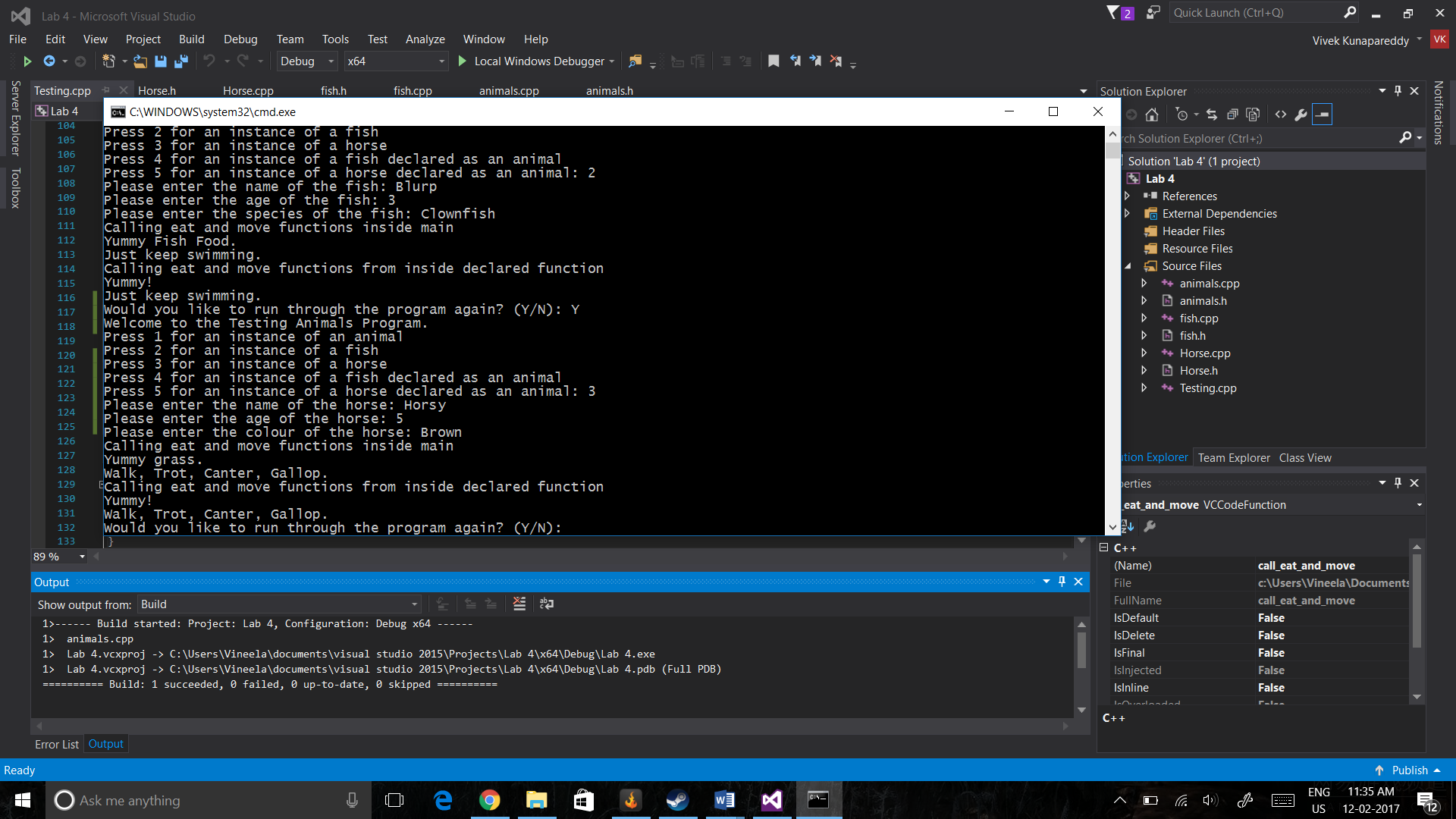
Fish and Horse derived from Animal but declared as Animal: They will have access as well to the public member functions and also to the private member variables. They are unable to override move() and eat() however.

**Task 3:**

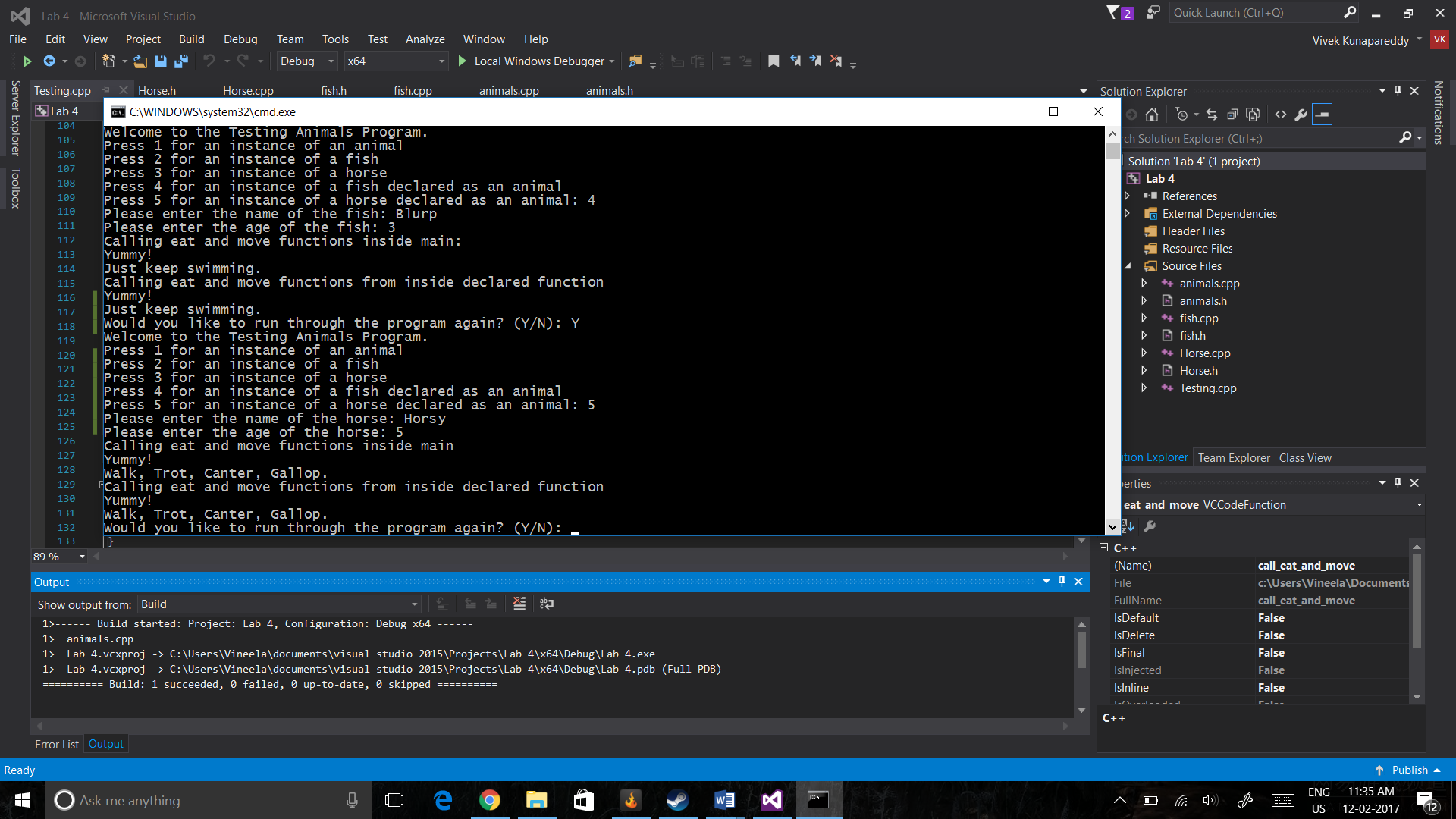
Screenshot with outputs for selections(1 & 2):



Screenshot with outputs for selection 3:



Screenshot with outputs for selection(4 & 5):



Discussion of results:

The output of the animal class was according to expectation.

The outputs of the fish and horse class declared as themselves were a bit off expectation. Specifically, the eat() function which was not declared as a virtual function was not overridden when passed into the function which only accepted animals as a parameter.

The outputs of the fish and horse declared as animals also showed this behaviour where the eat() function was never overridden but the move() function was always overridden.

The original assumptions were wrong because we assumed the non-virtual functions would always be overridden which is clearly not the functionality being displayed.

**Objectives explored:**

The objectives explored in this class were inheritance in an object oriented programming paradigm and how virtual functions behave in such a paradigm. This will be extremely important as software engineers as learning the fundamentals of object oriented programming will help when transitioning into the workforce.

**Group Contribution:**

Yuan designed and implemented the animals class.

Vivek designed and implemented fish and horse classes.

The rest of the testing program and debugging was done together