**Part 1 - Manager Functions**

1. Download the Inherit\_InLab.zip file from the assignment and extract it.
2. Load the solution into Visual Studio and review all associated files and classes.
3. Using the Elk class and associated files as a guideline make an Otter class that has a Boolean data member called m\_sea as well as all of the methods included with the Elk class.
4. Instantiate the following objects in main.
   1. Mammal m;
   2. Elk cow;
   3. Elk bull(3);
   4. Otter river(false);
   5. What is the output?

Mammal Default Ctor

Mammal One Arg Ctor

Elk Default Ctor

Mammal One Arg Ctor

Elk One Arg Ctor

Mammal One Arg Ctor

Otter One Arg Ctor

Otter Dtor

Mammal Dtor

Elk Dtor

Mammal Dtor

Elk Dtor

Mammal Dtor

Mammal Dtor

1. Call the "WhatAmI" method from each of the objects. What Additional output is displayed?

I am a Mammal

I am an Elk

I am an Elk

I am an Otter

1. Assign the "bull" object into "m". What additional output is displayed?

Mammal op=

1. Call the "WhatAmI" method from "m" again. What additional output is displayed?

I am a Mammal

1. Instantiate a new Elk object using bull to initialize it. What additional output is displayed?

Mammal One Arg Ctor

Elk Copy Ctor

Elk Dtor

Mammal Dtor

1. Change the object from the previous step to a Mammal object. What changes are there to the output?

Mammal Copy Ctor

Mammal Dtor

##### Part 2 - Virtual Functions and Polymorphism

1. Declare and array of 3 Mammal pointers.
   1. In element 0 dynamically allocate a Mammal.
   2. In element 1 dynamically allocate a Elk.
   3. In element 2 dynamically allocate a Otter.
2. Use a loop to iterate through the array calling the "WhatAmI" function from each element. What is displayed? Are the appropriate destructors called?

Mammal Default Ctor

Mammal One Arg Ctor

Elk Default Ctor

Mammal One Arg Ctor

Otter Default Ctor

I am a Mammal

I am a Mammal

I am a Mammal

Yes the dtors are called.

1. What is the size of a "Mammal"? What is the size of an "Elk"?

Both are 32.

1. Modify the Mammal class to make the "WhatAmI" method virtual. What is displayed now?

Mammal Default Ctor

Mammal One Arg Ctor

Elk Default Ctor

Mammal One Arg Ctor

Otter Default Ctor

I am a Mammal

I am an Elk

I am an Otter

1. What is the size of a "Mammal"? What is the size of an "Elk"?

Both are 32.

1. Make the destructor in "Mammal" virtual. What is displayed? Are the appropriate destructors called?

Mammal Default Ctor

Mammal One Arg Ctor

Elk Default Ctor

Mammal One Arg Ctor

Otter Default Ctor

I am a Mammal

I am an Elk

I am an Otter

The dtors were not called.

**Part 3 - Pure Virtual Functions and Abstract Base Classes**

1. Create a new class called "Animal".
   1. Move the m\_species data member from "Mammal" to "Animal".
   2. Make the only constructor for "Animal" a one argument constructor that takes a string and initializes the m\_species data member.
   3. Make Mammal publicly derived from Animal.
   4. Make the modifications necessary to the other classes to compile the project and get it working correctly.
2. What is the size of an "Animal"?

32

1. What is the size of a "Mammal"? What is the size of an "Elk"? Did they change size? If so, why?

Both are 32 neither changed size.

1. Change the stub file so that the array is an "Animal" array of pointers with 4 elements, the first being an "Animal". Compile the project. What happened and why?

Each element of the array said it was an Animal because the WhatAmI function is not virtual.

1. Modify the "Animal" class so there is a pure virtual function declaration for the "WhatAmI" method. Re-compile the project. What happened and why?

Could not compile because I cannot instantiate an abstract class.

1. Change the array so that there are no "Animal" objects. Re-compile and run the program. What is the output?

Animal One Arg Ctor

Mammal Default Ctor

Animal One Arg Ctor

Mammal One Arg Ctor

Elk Default Ctor

Animal One Arg Ctor

Mammal One Arg Ctor

Otter Default Ctor

I am a Mammal

I am an Elk

I am an Otter

1. Comment out all references to the "WhatAmI" method in the "Mammal" class. Re-compile the project. What happened and why?

I got an error cannot instantiate abstract class because without the WhatAmI in the Elk class it too becomes abstract.

1. Put the "WhatAmI" method back in the "Mammal" class. Re-compile the project.
2. Comment out all references to the "WhatAmI" method in the "Elk" class. Re-Compile the project. What happened and why?

When you get to the WhatAmI function for the Elk it says I am a Mammal since it had to call the Mammal version since Elk did not have one.

1. Put the "WhatAmI" method back in the "Elk" class. Re-compile the project.
2. Create a "Bird" class that is publicly derived from "Animal" and a "Duck" class that is publicly derived from "Bird". Make sure each class has the required functionality.
3. Modify the pointer array in main so that is has room for two more objects. Make the last two elements a "Bird" and a "Duck" respectively. Modify the loop so that it includes these two new elements. What is the output?

Animal One Arg Ctor

Mammal Default Ctor

Animal One Arg Ctor

Mammal One Arg Ctor

Elk Default Ctor

Animal One Arg Ctor

Mammal One Arg Ctor

Otter Default Ctor

Animal One Arg Ctor

Bird Ctor

Animal One Arg Ctor

Bird One Arg Ctor

Duck Ctor

I am a Mammal

I am an Elk

I am an Otter

I am a Bird

I am a Duck