Lab 5

Wednesday, September 1, 2021 6:43 PM

3.2

3 c) Virtual keyword is put in the Animal class' speak method and the Dog class would have its own Speak method with the override keyword to polymorph the method to the Dog specific one, output is as expected

```
virtual void speak()
{
    cout << "Animal speaks " << endl;
}
```

```
void speak() override
{
    cout << _name << " the dog woofs" << endl;
}
```

4. Animal is now unable to be declared as it is an abstract class now.

3.3

2. Without the virtual keyword the pointer will call the speak method from the main Animal class instead from Dog.

3c) All 3 speak functions will call the speak method from the Animal class. This is because they are all declared to be pointers or references to a Mammal object, which has no speak method of its own, so it uses the speak method from Animal. If a speak method were to be implemented in the Mammal class, that would be called instead.

4 b) Move all animals runs as expected, calling the methods within the scope of the Mammal class or higher (its superclass, Animal) rathe than the individual methods defined in each unique animal's class.

Changing the array from Mammal to animal will cause the program to fail for 2 reasons

1. The Animal class is an abstract class and cannot be declared as an animal object
2. The Animal class does not have the 'eat' method defined