Your Name

[Email address]

Questions of Assignment

**Question1:** **Try declaring the health member function private instead of public. Why do you get the resulting error?**

Ans: When we declare ‘heath member function’ private instead of public, it would not be accessible in main. Because private member functions are only accessible inside a class. While public members are accessible where we create it object. Therefore, it gives error of having undefined.

**Question2: Comment out the entire implementation of Pet::eat, all the way from void Pet::eat(int amt) to its close curly brace. Why do you get the resulting build error?**

Ans: when we comment out the entire implementation of Pet::eat, all the way from void Pet::eat(int amt) to its close curly brace, it gives build error. Because, it’s not anymore feds. The amount of health level is not going to increase. It will only use previous health levels which we provide when we made object by giving in constructor.

**Question 3:** **In main, replace myPets[0] = new Pet(“fluffy”, 2); with myPets[0] = new Pet(“fluffly’); or myPets[0] = new Pet; Why do you get the resulting compilation error?**

Ans: Because when we write ***myPets[0] = new Pet(“fluffly”);*** it give error due to not having constructor with this parameter. While constructor has two arguments and we are giving only one. So, it is giving error. In second statement error is due to same reason. It is not giving any aurgument while constructor need two arguments.

**Question4: Try removing the const from the implementation, but not the declaration of the alive member function. What happens? Why?**

Ans: When we remove const from the implementation of alive() member function it give an two error,

One is because, declaration and implementation not remain the same, so it gives error.

Second is because we are calling it through const object(which we pass as argument in function reportStatus). When we call non constant member function through const object it gives error.

**Question 5: Try removing the const from both implementation and declaration of alive member function , Explain why the use of that function doesn’t compile in reportStatus, but does compile in careFor and main**.

Ans: When we remove const from alive() member function it give an error, **In reportStatus()**

If we say why giving this error, it is because we are calling it through **const object**(which we pass as argument in function reportStatus). When we call non constant member function through const object it gives error.

**In careFor() and main()**

It does compile in careFor and main, because it is called by **non const objects**.