Code Analysis

**Given the following:**

*class Animal*

*{*

*public virtual string speak(int x) { return "silence"; }*

*}*

*class Cat : Animal*

*{*

*public string speak(int x) { return "meow"; }*

*}*

*class Dog : Animal*

*{*

*public string speak(short x) { return "bow-wow"; }*

*}*

**Question: Explain why the block below does not emit “bow-wow”:**

Animal d = new Dog();

Console.Write(d.speak(0));

This is due to a concept called method overriding and type checking in object oriented programming.

* The Dog class inherit from Animal class, Dog class has access to all methods of Animal class
* Dog class has speak(short x) instead of int which overrides the inherited method from Animal class
* When creating instance d of Animal class and assign a Dog object, compiler sees the d variable as Animal

Therefore, block does not emit "bow-wow" because the method speak(short x) in Dog is an overload, not an override. The method resolution is based on the type of the variable d (which is Animal) and the argument type (int). The Dog class does not override the speak(int x) method from the Animal class. Hence, the speak(int x) method from Animal is called, which returns "silence".

To get “bow-bow”, we need to cast d variable to Dog and then call speak method with short parameter (directly assign Dog reference (Dog d=new Dog())).

Dog d = new Dog();

Console.Write(d.speak((short)0));

Or

Animal d = new Dog();

Console.Write(((Dog)d).speak(0));

**Given the following:**

*class A*

*{*

*public int a { get; set; }*

*public int b { get; set; }*

*}*

*class B*

*{*

*public const A a;*

*public B() { a.a = 10; }*

*}*

*int main()*

*{*

*B b = new B();*

*Console.WriteLine("%d %d\n", b.a.a, b.a.b);*

*return 0;*

*}*

**Question: Outline any issues/concerns with the implemented code.**

The following code has some issues:

* Use of constant variable, it should be initialize at the time of their declaration. They can be only be values type eg. Int, string, bool, etc
* Instead of using const variable, we can use to readonly and initialize the readonly field.
* Syntax for console.writeline is incorrect, need to use c# string interpolation or format specifiers.

So, the corrected call method would be.

class A

{

public int a { get; set; }

public int b { get; set; }

}

class B

{

//public const A a;

public readonly A a;

public B()

{

//a.a = 10;

a = new A();

a.a = 10;

}

}

Int main()

{

B b = new B();

//Console.WriteLine("%d %d\n", b.a.a, b.a.b);

Console.WriteLine("{0} {1}",b.a.a, b.a.b);

}