# Prob 2 Write-up

**Explain how access modifiers constrain access to fields by child classes:**

* Public – field is directly accessibly through the object (never use!!)
* Private – field is NOT accessible through the object. Private fields can only be accessed by the class that defines them and are not accessible by child classes
* Protected – field is also NOT accessible through the object. However, the field is directly accessible from within the class it is defined in, or to code in child classes. It allows child classes access to fields defined in a parent class.

**How does the use of properties affect access?**

* Properties allow access to private and protected fields. They enforce encapsulation by allowing the fields to stay hidden, but allowing them to be modified and accessed by child classes.

**Explain your results on the constructor calling sequence demo:**

* When I execute main and create an instance of TigerCub, the console prints out the order of which constructors are called: first the Mammal constructor, then the Tiger constructor, and finally the TigerCub constructor.
* This illustrates the constructor calling sequence when you call a constructor linked to base class. The base class constructor executes first, and then on down the line.

**Find an example that justifies the use of the protected keyword on a field:**

* **protected** is useful when you want your class and all derived (child) classes to be able to access the method or variable, but you don't want it to be public.
* A good justification for using the protected modifier is when using aggregation – and you want derived classes to have access to fields in classes they derive from.