**Explain your naming scheme:**

* I used a hierarchy of mammals for my naming scheme, starting with the general animal family “mammals” (MammalClass) – since the base class is usually general, while child classes are more specific. From there, I derived two species – Tiger and Bear – and from those I derived two subspecies from each, creating a full hierarchy.
* MammalClass is my abstract base class
* From there, TigerClass and BearClass are derived from the base class
* From TigerClass, I derived SiberianTigerClass and BengalTigerClass, and from BearClass, I derived BrownBearClass and GrizzlyBearClass.
* This is a hierarchy that is 3 deep, and 2 wide at the second level

**Describe what tool views/fields you had to use to create your class hierarchy:**

I took some advice from Kyle on Slack and now use CTRL+ALT+X to get to my Toolbox.

The Toolbox is where you go to build a class hierarchy using ONLY the class diagram tool.

First, you right click your project 🡪 Add New Item 🡪 Class Diagram

Next, you open the ToolBox using CTRL+ALT+X

Then, to create a class, you can click and drag “class” from the Toolbox into the diagram. You repeat this process for the remaining classes.

After that, you need to add inheritance to complete a class hierarchy. To do this, you click on “Inheritance” in the Toolbox. Next, click the class you want to be the inheritor, and right after that click on the class you want to inherit *from*. If you performed these steps correctly, you should see and arrow showing the inheritance.

**Describe the use of the “abstract” keyword. What constraints does it put on the class that it modifies? What affect does it have on class members (like fields and methods)?**

* Because instantiation on a general (base) class often doesn’t make sense, the abstract keyword is used to prevent instantiation. The abstract base class still has the same role as a regular base class – to define common data and behavior – it just prevents instantiation.
* Using the abstract keyword on a method in the base class is the mechanism used to ensure that all child (inherited) classes implement the method. It forces implementation. The child class must use the override keyword to implement an abstract method. It allows you to define a method signature in the base class without coding an implementation.
* The purpose of the abstract method is to define behavior common to all derived classes, but rely on the derived classes to implement the details (like a contract).

**Describe the code generated. How does this help you code?**

* The instructions just say to create a class diagram, so I wasn’t sure if you wanted any actual code, but class diagrams help you code by seeing the structure behind the code in a more visual, human-friendly way.