NOTE: Problem 1 Kyle Writeup (he/we assigned and completed this before the requirements for the assignment were changed)

* Explain your naming scheme

My naming scheme is essentially three tiers of a common character generator found in numerous games. Starting with a base character, the user can choose to be a human or an orc, or undead. From this choice they can choose their class of character, warrior, priest, shaman, paladin, berserker, or druid.

* Describe what tool views/fields you had to use to create your class hierarchy. In other words, explain the use of the class diagramming tool to someone who hasn't ever used it before.

I simply used the class diagram and its associated tool box (ctrl+alt+x). From here an abstract class can be created from the abstract class selection, then standard classes can be inserted as the user pleases. Inheritance can be implemented by drawing lines from the desired parent class to the desired child using the “inheritance” selection from the tool box.

* 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)?

The abstract keyword is used when the programmer would like to hide the internal details of a class, method, property, indexer or event only showcasing its functionality. Classes that are marked as abstract must be implemented from a non-abstract class that inherits from it. In terms of an abstract method, these can only be implemented inside an abstract class, they are also implicitly virtual methods therefore the user does not need to write public abstract virtual void myMethod. Fields cannot be abstract; however, they can be contained in an abstract class and, depending on their protection level, passed to child classes or not. With an abstract property, the accessor will not be provided by a property declaration.

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

The code and classes generated from this practice allow users to easily establish methods, properties, fields and constructors from the diagram screen. Afterwards they can look at the actual program to see the code that has actually been written.