**Abstraction and Inheritance (50pts)**

**Project Description & Goal**

The goal of this project is to practice abstraction and inheritance. This is where the “Design Thinking” of “Coding and Design Thinking” comes in to play. By creating an elegant hierarchy, your code becomes easy to read and use. Your code also becomes more dynamic.

**Project Specifications**

* Create an inheritance diagram of something of your choice.
* Codify your inheritance diagram.
* In Program.Main() write some code that demonstrates the elegance of your hierarchy.
  + Demonstrate that every item of a certain type can be unique, while also maintaining similarities with objects of other types.

**Project Learning Objectives**

* Understand abstraction
* Understand inheritance
* Think creatively about how to design an effective hierarchy of inheritance.

**Project Demonstrated Competencies**

1. Inheritance diagram organizes fields/methods/classes in a logical way.
2. Project implements inheritance diagram exactly as written.
3. Program.Main() uses these classes effectively to demonstrate the benefit of abstraction and inheritance.

**Rubric**

|  |  |  |
| --- | --- | --- |
|  | **Description of perfect implementation** | **Score** |
| Competency #1 | 2 levels = 80%, 3 levels = 100%. Little to no instances of multiple classes duplicating a variable or method. | \_\_\_  20 |
| Competency #2 | Implement the code exactly as you drew the diagram. If you realize you need to make changes to the code, make changes to the diagram. | \_\_\_  20 |
| Competency #3 | Program.Main() shows exactly why your diagram was effective. | \_\_\_  10 |