Cody Martin/Regina Pilipchuk

TASK: Programming Assignment 3 - Inheritance Behavior

Probs 4-5

**Problem #4**

**Problem Statement:** Create a hierarchy of classes that demonstrates polymorphic behavior

**Region 1:** Class hierarchy created using a Base class and a Child Class. Region 1 creates an array initialized with a member of each class in the hierarchy

**Region 2:** Here we create a child object “c1” and have it call the base method PrintBase(); sending its own parameters in (“Hi from the child class but calling a base method!”)

**Region 3:** Here we have c1 override the PrintOverride(); method within the child class.

**Region 4:** Her we use the “IS” keyword to see if c1 is of the **child** type using an if/else statement

**Region 5:** Her we use the “IS” keyword to see if c1 is of the **base** type using an if/else statement

**Region 6:** Here we try to convert the child type (c1) to base type using the “AS” keyword.

**Region 7:** Here we iterate over the object instances in the original array and print "iterating over the array with a for loop" per instance.

**Region 8:** Here we call a method per instance in the original array using a foreach statement.

**Region 9:** Here we use a switch statement to show it responding differently bases on object type in the original array

**Region 10:** Here we create an object of the “object” type and initialize it to the “base” class type. Then we cast it to the actual type so we can call the method. Next, we create an object of the base class but initialize it to a child type. Then we call a specialized method SpecializedMethod();.

**Problem #5**

Here we create a class with two fields. The class name is “Animal” and the two fields are a string type names “name” and an int type name “age.” We use these two different data types to show the sorting capabilities we can achieve. First, we initialize 6 instances and provide 1 int and 1 string parameter. Then we insert these instances into an array. Then we sort and print. In the class Animal, we implement the “IComparable” interface to show comparison capabilities.