Name: Torin Tashima

Partner: Katrina Voll-Taylor

Write-Up Problem 4 – Polymorphic Behavior Demos

This write-up is written to explain the use of polymorphism following these steps in the coding portion:

* An array of the base class type is created and initialized with a member of each class in the hierarchy
* Show that a method belonging to the base class is called from a child instance
* Demonstrate overriding a base class method by a child
* Demonstrate a child method calling the parent's method
* Demonstrate use of "as" keyword
* Demonstrate use of "is" keyword
* Demonstrate calling a method in a for loop iterating over the instances in the array
* Demonstrate calling a method in a foreach loop
* Demonstrate switch statement that responds differently depending upon the type of the object
* Create an object of "object" type, but initialized of the base class type, then cast it to the actual type so you can call a method
* Create an object of the base class type but initialized to a child type that has a specialized method. Call the specialized method by using casting.

Classes used in hierarchy: Base (parent class), Child (child class)

**Create and initialize array of the base class type with a member of each class in the hierarchy:**

Under region “array of every member of classes in hierarchy”, an array of the Base class type is named ObjectArray and is initialized with a new Base object and a new Child object.

**Demonstrate method belonging to the base class being called from a child instance: \***

Region “base class method called from child object” iterates through the ObjectArray using a for loop and calls BaseMethod from each object. Since BaseMethod is never overridden by the Child class, both the Base and Child objects call the BaseMethod from the Base class.

**Demonstrate overriding a base class method by a child: \*\***

Region “demonstrate method overriding” iterates through the ObjectArray using a foreach loop, “for each Base Obj in the ObjectArray,” and calls PrintClass from each object. The PrintClass method is overridden by the Child class, so the Base object would print “I’m a base object!” while the Child object would print “I’m a child object. Goo.”

**Demonstrate a child method calling the parent’s method:**

Region “demonstrate child method calling parent’s method” iterates through the ObjectArray and calls ParentsMethod from each object. The ParentsMethod is overridden by the Child class, but uses the “base” keyword to call ParentsMethod from its parent or base method. The Base object would print out “Base.ParentsMethod” while the Child object would print out both “Base.ParentsMethod” and “Child.ParentsMethod” as a result.

**Demonstrate use of “as” keyword:**

Region “demonstrate use of ‘as’ keyword” iterates through ObjectArray but each object, or Obj, through the iteration now has the Object class type instead of the Base class type. The “as” keyword converts the objects with the Object class type into an object with the Base class type, as shown on the line “Base b = Obj as Base;”

**Demonstrate use of “is” keyword:**

Region “demonstrate use of ‘is’ keyword” iterates through ObjectArray and contains an if/else if statement. If the object is a Child, then it would print “Child” to the console. Else if the object is a Base, then it would print “Base” instead.

**Demonstrate calling a method in a for loop iterating over the instances in the array:**

See “Demonstrate method belonging to the base class…” section marked with \*.

**Demonstrate calling a method in a foreach loop:**

See “Demonstrate overriding a base class method…” section marked with \*\*.

**Demonstrate switch statement that responds differently depending upon the type of the object:**

Region “switch statement” iterates through ObjectArray and contains a switch statement taking in Obj as a parameter. In case Obj is a Child, it prints “switch Child” to the console. If Obj is a Base, it prints “switch Base” and if Obj is neither of those, the default switch case is called and prints “switch default.”

**Create an object of “object” type, but initialized of the base class type, then cast it to the actual type so you can call a method:**

Region “create object of object type and initialize to base type” starts with an initialization of an Object named newObject assigned to a value of a new Base object. To access the PrintClass method in the Base class, the Base word is used in parentheses preceding the object name newObject to cast the object as a Base class type instead of an Object class type so it can therefore access the method.

**Create an object of the base class type but initialized to a child type that has a specialized method. Call the specialized method by using casting:**

Region “create object of base type and initialize to child type” is similar to the above section, but the Base object bassGuitar is initialized to a value of a new Child object. The Child word is used in parentheses preceding the object name bassGuitar to cast the object as a Child class type so the PrintClass method will print out “I’m a child object. Goo.” Instead of “I’m a base object!”