Carlos Solorio

Assignment 3 – Ex3

Questions

1. (Programming in the General) How does polymorphism enable you to program “in the

general” rather than “in the specific”? Discuss the key advantages of programming “in the general.” (2 pts)

In object oriented programming, polymorphism allows us to focus on operations that are useable in all classes via a hierarchy. We can take code from each one of these classes and use them so long as the general “base” code is used. These “base” guidelines allow us to add them to new classes without making any modifications. This allows us to essentially re-use a lot of code during programming to shorten the writing process.

2. (Abstract Methods) What are abstract methods? Describe the circumstances in which an abstract

method would be appropriate. (2 pts)

Abstract methods are basically a *virtual method* that can only be declared in abstract classes (keyword abstract). Because of this, they cannot be implemented into your basic code, they can only be used within he virtual method. You would likely implement an abstract method when a method for a superclass would not be useful.

3. (Assigning Base Class and Derived Class References) Discuss ways (there are 4 ways) in which you can

assign base-class and derived-class references to variables of base-class and derived-class types. (4 points)

“Skipping for now”

4. (Abstract Classes vs. Interfaces) Compare and contrast abstract classes and interfaces. Why

would you use an abstract class? Why would you use an interface? (2 points)

Abstract classes allow you to provide functions to the subclasses, however their implementation is much more useful when you plan on making changes throughout the life of the program. When you make a change in the abstract class, all inheriting classes will have the new functionality without problems. As for interfaces, they are often more used when the functionality of a class is simpler and more “permanent”. When the possibilities of classes and subclasses are lower, interfaces are much better to use, mainly because the possibilities of classes and subclasses in the abstract hierarchy would be much more work than necessary.