For this assignment, I was instructed to create a testing scenario to compare the results of abstraction versus interface implementation. Both of these concepts are handy when organizing code. Abstract classes are a median between interface implementation and classes. Whereas interface implementations solely contain method signatures. Abstract classes contain the method body and signatures. The methods can be overridden in the child class. The interfaces used in the assignment were Drawable, Calculable, and Describable. They all held methods that were implemented when used. There is more flexibility in using interfaces because you can implement the same method, but create a different body, providing more opportunities for change in code. This could also be proven to be shifty because the coder and/or user could be confused by the method having two different implementations and/or meanings. Abstract classes are more reusable and easier to read because they provide more stability. The stability is provided in the method body always being provided. Without the method body, the code can become convoluted. The abstract class method scales better than the interface implementation because it provides information, making it easier for the user. I would use the abstract class in a real-world model because it allows less room for error.