Assessment Solutions

Chapter 8 – Mastering Abstract Classes

1. **a – d**: Please see `Chapter08/Assessments/Chp8-Q1.cpp` in the GitHub repository.

   e: Depending on your implementation, your `Shape` class may or may not be considered an interface class. If your implementation is an abstract class that contains no data members and only abstract methods (pure virtual functions), your `Shape` implementation is considered an interface class. If your `Shape` class, however, stores `area` as a data member once it has been calculated by the overridden `Area()` method in the derived classes, it is then just an abstract base class.