Assessment Solutions

Chapter 16 – Using the Observer Pattern

1. **a – b**: Please see `Assessments/Chapter16/Chp16-Q1.cpp` in the GitHub repository.
2. Other examples which may easily incorporate the Observer pattern include any application requiring customers to receive a notification of backordered products that they desire. For example, many people may wish to receive the Covid-19 vaccine and wish to be on a waiting list at a vaccine distribution site. Here, a `VaccineDistributionSite` (the subject of interest) can be inherited from `Subject` and contain a list of `Person` objects, where `Person` inherits from `Observer`. The `Person` objects will contain a pointer to the `VaccineDistributionSite`. Once enough supply for the vaccine exists at a given `VaccineDistributionSite` (that is, a distribution event has occurred), `Notify()` can be called to update the `Observer` instances (people on the wait-list). Each `Observer` will be sent an `Update()`, which will be the means to allow that person to schedule an appointment. If the `Update()` returns success and the person has been scheduled for an appointment, the `Observer` can release itself from the waiting list with the `Subject`.