Assessments

The programming solutions for each chapters' questions can be found in our GitHub repository at the following URL: `https://github.com/PacktPublishing/Demystified-Object-Oriented-Programming-with-CPP/tree/master`. Each full program solution can be found in the GitHub under the appropriate chapter heading (subdirectory, such as `Chapter01`) in the subdirectory `Assessments`, in a file that corresponds to the chapter number, followed by a dash, followed by the solution number in the chapter at hand. For example, the solution for question 3 in chapter 1 can be found in the subdirectory `Chapter01/Assessments` in a file named `Chp1-Q3.cpp` under the aforementioned GitHub directory.

The written responses for non-programming questions can be found in the following sections. Should an exercise have a programming portion and a follow-up question, the answer to the follow-up question may be found both in the next sections and in a comment at the top of the programming solution on GitHub (as it may be appropriate to review the solution in order to fully understand the answer to the question).

Chapter 16 – Using the Observer Pattern

1. **a – b**: Please see `Chapter16/Assessments/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`.