

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 19 – Using the Singleton Pattern

1. **a – c:** Please see `Chapter19/Assessments/Chp19-Q1.cpp`
2. We cannot label the `static instance()` method as `virtual` in `Singleton` and override it in `President` simply because static methods can never be `virtual`. They are statically bound and also never receive a `this` pointer. Also, the signature may need to be different (and no one likes an un-intentional function hiding situation).
3. Other examples which may easily incorporate the Singleton pattern include creating a Singleton CEO of a company or a Singleton TreasuryDepartment for a country or a Singleton Queen of a nation. Each of these Singleton instances offers the opportunity to establish a registry to keep track of multiple Singleton objects. That is, many countries may have a single Queen. In this case, the registry would allow not just one Singleton per object type, but one Singleton per other qualifiers, such as *nation*. This is an example of the rare case where more than one Singleton object of a given type can occur (but always a controlled number of such objects).