Lab 4

Course: CSE 165

All the exercises below are selected from the textbook: Thinking in C++ (volume 1).

1. [Exercise-4 on Page 505] Write a function that takes a pointer argument, modifies what the pointer points to, and then returns the destination of the pointer as a reference. [20 points]

2. [Exercise-7 on Page 506] Create a function that takes an argument of a reference to a pointer to a pointer and modifies that argument. In main( ), call the function. [30 points]

3. [Exercise-27 on Page 658] This exercise creates the design pattern called proxy. Start with a base class Subject and give it three functions: f(), g(), and h(). Now inherit a class Proxy and two classes Implementation1 and Implementation2 from Subject. Proxy should contain a pointer to a Subject, and all the member functions for Proxy should just turn around and make the same calls through the Subject pointer. The Proxy constructor takes a pointer to a Subject that is installed in the Proxy (usually by the constructor). In main(), create two different Proxy objects that use the two different implementations. Now modify Proxy so that you can dynamically change implementations. [50 points]

Requirements:

* Usage of spaces, blank lines, indention, and comments for readability.

* Descriptive names of variables, functions, structs, classes, and objects (if any).

* Appropriate usage of structs, classes, and objects (if any).

Penalties:

* Zero if you have possession of a copy of online solutions or work done by someone else.

* 5-point deduction per day late