Name:

Proxy Pattern

CSC-490, Winter 2020

1. (a) (2pts) Copy down the intent of the Proxy pattern from the Gang of Four:

   (b) (5pts) Draw the class diagram for the Proxy pattern.

   (c) In the *Applicability* section for the Proxy pattern, the Gang of Four describe several types of proxies. Write the description of each of the following:

      i. (2pts) Remote proxy:

      ii. (2pts) Virtual proxy:

      iii. (2pts) Protection proxy:

   (d) (3pts) Which of the following type of proxies (remote, virtual, protection) is always responsible for creating objects? Draw the modified class diagram for that type of proxy with the appropriate creates-a arrow.

2. Consider an abstract class `DataProcessor` with a method `getData()`. Each concrete subclass is responsible for computing a different type of data from a file. These concrete classes have constructors of the form `ConcreteDataProcessor(file)`. When that constructor is called an algorithm is performed on the given file to extract the desired data, and that desired data is stored in a data member. Depending on the algorithm and the size of the file, the instantiation process for `ConcreteDataProcessor` might be very expensive.

   Suppose you are working on a program that involves lots of concrete `DataProcessor` objects. In particular, your program (for whatever reason) needs to pass these objects around all the time. However, the number of times `getData()` is called is entirely determined at run time. In fact, your program could be used without ever calling `getData()` on many of its `ConcreteDataProcessor` objects.

   Your program is running slow due to the instantiation of all the concrete `DataProcessor` objects. You decide to use a proxy to speed things up.

   (a) (2pts) Which type of proxy (remote, virtual, protection) is appropriate for this situation?

   (b) (6pts) Draw the class diagram that illustrates the use of the Proxy pattern in this story.

   (c) (2pts) The Proxy should have just one data member. What type is that data member?

   (d) (4pts) What should the Proxy's data member be set to when the Proxy is instantiated? Why?

   (e) (6pts) Write detailed pseudocode for the method `getData()` in the Proxy.

3. Consider a model for a (simplified) video streaming application with the following structure:

   - There is an abstract class `Film` with a method `play()`. There is also a concrete subclass `ConcreteFilm` which has a data member `filmID`. The constructor for `ConcreteFilm` takes a `filmID` as a parameter.
   - There is a class `User` with a method `watch(filmID)`. When `watch(filmID)` is called, the appropriate film object should be created and then played. There are concrete subclasses of `User` named `AdultUser` and `ChildUser`.
   - The `ChildUser` is restricted from watching certain films. In order to enforce this restriction, there is a `ChildFilmProxy` that holds the list of restricted filmID's.

   (a) (8pts) Design a program for this story that (1) uses the Factory Method pattern to implement `watch(filmID)`, and (2) uses the Proxy pattern to implement the `ChildFilmProxy`. Draw the corresponding class diagram.

   (b) Include detailed pseudocode for each of the following:

      i. (2pts) `watch(filmID)` in `User`.

      ii. (4pts) The factory methods in the concrete user classes.

      iii. (4pts) `play()` in `ChildFilmProxy`

4. A remote proxy:

   (a) (5pts) Describe a situation where a client needs to have access to a remote object (i.e. an object in a different address space). For instance, your example could require a client to access an object via some network. Your example should include at least one method in the remote class that the client needs to call.

   (b) (5pts) Draw a class diagram for your situation where the client uses a proxy to access the remote object.

   (c) (5pts) Explain the benefit of using a proxy as opposed to having the client access the remote object directly. In particular, what responsibility is the proxy encapsulating that would otherwise be the client's responsibility?

5. (4pts) We now have several patterns that are sometimes called wrappers. Write the name of the design pattern which uses a wrapper with the following *intent*.

   (a) Wraps another object and provides a different interface to it.

   (b) Wraps another object and provides additional behavior for it.

   (c) Wraps another object to control access to it.

   (d) Wraps a bunch of objects to simplify their interface.