Answer each of the questions below, demonstrating a full understanding of the concepts involved in each question. While you may discuss the questions with your classmates, your responses must be in your own words and your designs must be your own.

1. (10 points) One criticism of the Chain of Responsibility Pattern is that the decision-making associated with chain execution, which is not the business of the subclasses in the chain, is tightly coupled with the request-handling associated with chain execution, which is the business of the subclasses in the chain. Outline circumstances in which loose coupling of these two tasks would be necessary.

2. (5 points each) The Command design pattern, illustrated below, encapsulates requests as objects, facilitating the parameterization of requests.

A company has set up a network crisis center to enable it to quickly respond to problems that may arise on any of its dozens of Web servers. For example, if a major problem arises at its Denver server, someone at the crisis center might need to shut down that server by connecting to it, issuing multiple shutdown commands (backing up files, rerouting network connectivity around the server, performing system diagnostics, etc.), and ultimately disconnecting.
Similarly, if a less severe problem arises at the Miami server, crisis center personnel might need to reboot that server by connecting to it, issuing multiple reboot commands (transmitting warnings to connected servers, loading appropriate boot code, confirming proper restart, etc.), and then disconnecting.

Currently, crisis center personnel must issue long sequences of commands to one of many connected servers whenever a problem arises. This could easily lead to commands being issued in the wrong sequence or to the wrong destination. By encapsulating the individual commands needed to handle a particular problem into a single Command object and then binding that object to a specific receiver, the potential for miscommunication is greatly reduced.

a) Modify the UML class diagram above to illustrate how the Command design pattern could be used to handle the Denver shutdown scenario and the Miami reboot scenario. Use a separate diagram for each scenario, and clearly identify the classes that would be involved in each specific scenario.

b) The Command design pattern can be modified so it would be possible to undo the last several commands that were executed. Alter the provided class diagram and develop a new UML sequence diagram to illustrate how this modification can be effected.

3. (5 points) A simple blog can be implemented by means of the Interpreter Pattern, with the expressions being the simple insert, update, and delete commands that a user might use when posting, editing, or removing an entry from the blog. Under these circumstances, specify what the TerminalExpression, CompoundExpression, and Context components of the pattern consist of.

4. (10 points) A hash table has no inherent ordering, making concepts of its “first”, “last”, “current”, and “next” item meaningless. Explain whether this disqualifies it as a possible concrete aggregate in the Iterator pattern, illustrated at right.

5. (5 points) Notice that the FactoryMethod pattern is used by the CreateIterator function in the Iterator pattern to ensure that the correct type of ConcreteIterator is returned to the ConcreteAggregate. Explain the advantages that the use of the Factory Method pattern affords under these circumstances.

6. (10 points) Modern network protocols use message-passing techniques that require each message to pass through several internal network devices on its way from its source to its destination. At any point in this process, the message has a set of headers and trailers that identify the network protocol layer at which the message is currently operating (e.g., the end user’s Application Layer, the specially encrypted Presentation Layer, the pathfinding Network Layer, etc.). Each device may alter the message (usually by modifying the various headers and trailers), simply forward the message, or even discard the message (if some serious problem is detected). Specify which design pattern corresponds to this means of handling messages. Explain your response.

This assignment is due on your drop-box by 9 AM on Thursday, March 26, 2009.