Project Report:

You should include a report describing your critical design choices and the reasoning behind them. This should include a description of changes made between the meeting with me and the final submission of the project. Some things to think about (both when writing about your design and doing it):

• Which classes need to know that there are multiple strategies?

The ElectionManagerFactory method needs to know about the multiple strategies. No other classes need to know about this because the user passes the strategy ID to the factory, the factory checks the validity of the ID and decides what to do with it by creating an instance of a concrete strategy according to the ID passed.

• Which class is responsible for deciding which strategy should be used?

The factory is responsible for deciding which strategy should be used. The user gives the factory the ID for the strategy he/she wants to use. The factory then creates an instance of a concrete strategy according to the ID passed.

• Which class is responsible for keeping track of how many points a candidate has?

The CandidateData class is responsible. This class holds the information about each candidate. The Election class then holds an array of CandidateData which represents the Voting.csv file in a more workable format.

• Which class is responsible for determining the winner?

The Election class is responsible for determining the winner. The Election class holds an attribute of Strategy

according to the Strategy design pattern. Each concrete Strategy class has its own unique method of calculating the points for each candidates given. The Election class simply takes the candidate with the most points and calls it the winner.

• How does the Factory pattern encapsulate design knowledge so the main method does not need to know about some of the other classes?

The factory method is the one that checks for the validity of the strategy ID the user has chosen and determines which concrete strategy to instantiate. The user does not need to do the instantiation of these strategies but the factory method handles that.

• How much code did you have to modify to add your fourth strategy?

1. Create a new class that implements the Strategy interface

2. Write the getPoints() method for the new Strategy class

3. Add new if statement in the Factory class

• Which classes are the boundary classes? Entity? Control?

Boundary: IO, ElectionManagerFactory, ElectionManager

Control: Election, Strategy Classes

Entity: CandidateData

• How did you map the candidate name (from the votes file) to the candidate’s data (points) stored in your program?

I created a CandidateData class that holds the information about a candidate including his/her name and votes. The Election class holds an array of these CandidateData classes which is just a more workable version of the Voting.csv file data.