Project 3 Design Document

**Program Requirements:**

This program is meant to determine the winner of a series of elections. The program will open a text file that has been preformatted with the election results. We know that the text file will have one ballot per line with each value representing a candidate. We know that for each election, there will only be two candidates.

**Program Outputs:**

* Temporary variable
  + int temp = 0
  + Temporarily stores candidate indicator value to be compared.
  + Must be between the values [1,8]
* Counting variable
  + int counter = 0
  + Keeps track of the total number of votes.
  + Must be a positive, non-zero value
* Number variables
  + int num1 … num8 = 0
  + keeps track of each candidate’s individual ballot count.
  + Must be a positive values.

**Test Plan:**

The tests will verify that the program will display the data correctly. The test will also ensure that the program will work with a variety of different data sets. The winner of each election should be printed first and have the word “WINNER” printed on the same line. The loser of the race should appear second before moving on to the next election.

* Case 1: votesimple.txt
  + Expected Output:

Presidential Election Winner:

Cedric Diggory - 3 votes - WINNER (60%)

Harry Potter - 2 votes - (40%)

Vice Presidential Election Winner:

Vincent Crabbe - 4 votes - WINNER (80%)

Hermione Granger - 1 votes - (20%)

Secretary Election Winner:

Luna Lovegood - 3 votes - WINNER (60%)

Susan Bones - 2 votes - (40%)

Treasurer Election Winner:

Draco Malfoy - 4 votes - WINNER (80%)

Neville Longbottom - 1 votes - (20%)

* Case 2: votes1.txt
  + Example Output:

Presidential Election Winner:

Cedric Diggory - 5784 votes - WINNER (50.2957%)

Harry Potter - 5716 votes (49.7043%)

Vice Presidential Election Winner:

Hermione Granger - 8253 votes - WINNER (71.7652%)

Vincent Crabbe - 3247 votes (28.2348%)

Secretary Election Winner:

Luna Lovegood - 8281 votes - WINNER (72.0087%)

Susan Bones - 3219 votes (27.9913%)

Treasurer Election Winner:

Neville Longbottom - 5770 votes - WINNER (50.1739%)

Draco Malfoy - 5730 votes (49.8261%)

* Case 2: votes2.txt
  + Example Output:

Presidential Election Winner:

Harry Potter - 7634 votes - WINNER (76.34%)

Cedric Diggory - 2366 votes (23.66%)

Vice Presidential Election Winner:

Hermione Granger - 7739 votes - WINNER (77.39%)

Vincent Crabbe - 2261 votes (22.61%)

Secretary Election Winner:

Luna Lovegood - 7867 votes - WINNER (78.67%)

Susan Bones - 2133 votes (21.33%)

Treasurer Election Winner:

Draco Malfoy - 7732 votes – WINNER (77.32%)

Neville Longbottom - 2268 votes (22.68%)

**Solution Overview:**

We will be using the ‘iostream’ library to define our I/O stream objects. We will also be using the ‘fstream’ library to access the needed files. We start with creating our variables in the main function. An integer will be used to hold a temporary value (temp). An integer will also be used to store a counting value (counter). Lastly, the program uses 8 additional integer variables to represent the number of votes each candidate received (num1 … num8).

An ifstream object should be created to access the file. Create a function for sorting votes. The values of each election will be temporarily stored into temp. The function will determine which candidate the value corresponds to and adds a vote to the respective candidate. The function will be called repeatedly until the ‘eof’ is reached.

When ‘eof’ is reached, the program will use another function to determine the winner of each election. This is done by comparing the number of votes each candidate has and will the winner and loser in the appropriate format. The function will be called 4 times to display the results of each election. The function will also call another function which calculates the percentage of the popular vote and returns that value as an integer. This function will be called 8 times in total.

After the results are calculated, the file containing the election data should be closed and the program should return 0.

**Algorithm Flowchart**

