|  |  |  |
| --- | --- | --- |
| C:\Users\kedwards\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\A4UZLVLQ\MPj03848740000[1].jpg | CS151 – Programming Assignment 2  Combating Election Fraud | C:\Users\kedwards\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\GAQ4JGBS\MCj03013940000[1].wmf |

To combat election fraud, the county of Hawaii is instituting a new voting procedure. The ballot has a letter associated with every selection a voter can make. For example, consider the following sample ballot:

1. Mayor[[1]](#footnote-1)

A. Rob Ford

B. Billy Kenoi

2. County Council

C. Valerie T. Poindexter

D. Aaron Chung

E. Jen Ruggles

2. Proposition 17 - Monorail Taxes

F. Yes

G. No

3. Measure 1 – County Park Funding

H. Yes

I. No

4. Measure 2 – By Law Changes

J. Yes

K. No

After submitting their ballot, every voter receives a receipt that has a unique ID number and a record of their voting selections. For example, a voter that submits a ballot for Billy Kenoi, Jen Ruggles, Yes on Proposition 17, No on Measure 1, and Yes on Measure 2 might get a receipt with

ID 4925 : BDFIJ

The next day, the county posts all the votes on their web page sorted by ID number. This allows a voter to confirm their submission and allows anyone to count the vote totals for themselves. A sample list for the sample ballot is shown below:

|  |  |
| --- | --- |
| ID | VOTE |
| 4925 | BDFIJ |
| 4926 | ACFHJ |
| 4928 | BCGHK |
| 4929 | ADGHJ |
| 4930 | BEGIK |

In this assignment, you will write a program that will read the ballot and the posted list from separate files.

The ballot file will be structured in the following manner. The first line of the file will contain the number of categories. For each category, the file will contain the number of choices for that category, the name of the category and the choices for that category. The end of the ballot information will be marked by a 0.

The next file will contain the votes for the election. Each line will contain a voter id followed by a recordation of the votes.

In terms of structure, your program should have a class named VOTER that stores an individual’s voting record. At a minimum, the class should have the following components:

1. A dynamic array to store the votes for the user.
2. A Voter ID
3. A default constructor (you may assume there will be no more than 100 questions per ballot)
4. An overloaded constructor that takes the number of votes and a string of votes (e.g. “ACFHJ”)
5. Accessor functions for the member variables
6. Mutator functions for the member variables
7. A copy constructor
8. A destructor
9. An overloaded = operator

Your program can store all the voters for an election in either a dynamic array or a vector inside of the main program.

In terms of processing, your program should iterate over your chosen data structure to compute the results for the election.

In particular, your program should output the choices to a text file for each ballot item, the number of votes received by each choice for that ballot item, and the percentage (to 2 decimal places) for each ballot item. Obviously, this output should be tab delimited.

## Step 1 – Understand the Problem (5 Points)

Tally the votes from the sample ballot on page 1 to fill in the following tables. Submit the results for your election along with your program plan.

Mayoral Election

|  |  |  |
| --- | --- | --- |
| Choice | Number of Votes | Percentage |
| Rob Ford |  |  |
| Billy Kenoi |  |  |

County Council

|  |  |  |
| --- | --- | --- |
| Choice | Number of Votes | Percentage |
| Valerie T. Poindexter |  |  |
| Aaron Chung |  |  |
| Jen Ruggles |  |  |

Proposition 17

|  |  |  |
| --- | --- | --- |
| Choice | Number of Votes | Percentage |
| Yes |  |  |
| No |  |  |

Measure 1

|  |  |  |
| --- | --- | --- |
| Choice | Number of Votes | Percentage |
| Yes |  |  |
| No |  |  |

Measure 2

|  |  |  |
| --- | --- | --- |
| Choice | Number of Votes | Percentage |
| Yes |  |  |
| No |  |  |

## Step 2 – Create a Plan (20 Points)

Your program plan should consist of the following items.

1. A description of your program design that indicates how you will design your program to meet the assignment requirements. You should indicate:
   1. The **variables and classes** that your program will use and how you will name and use them.
   2. The **major functions** your program will have along with descriptions of their parameters.
   3. Descriptions of the **algorithms** your program will use.

**Note:** Don’t be afraid to use visual representations and diagrams such as UML for classes.

1. Your program design description should also indicate how your solution will be easily understood by the user and maintained by future coders.
2. Format your plan into a PowerPoint presentation in case you are called upon to present it at the beginning of class.

**Submit parts 1 & 2 by September 22, 2017.**

## Step 3 – Implement (60 Points)

Implement your program using separate compilation for the voter class and the main program. As always, your code should have appropriate comments to explain the program and the code. It should be written in a consistent and readable form and should compile without errors or warnings.

**Submit your code by September 29, 2017.**

## Step 4 – Test (15 Points)

Identify at least four tests that you want to run on your program. Provide a table with the following attributes:

* Test Number – A unique identifier for the test
* Test Description – A description of the test
* Expected Results – What the program should do
* Pass/Fail – A column to indicate whether it passed or failed
* Comments – Any comment on the test results.

**Hint:** You may want to construct test files with different initial conditions, although your program should work with the files provided with this assignment.

**Submit your test plans by September 29, 2017 with your program.**

1. This will never happen, but this is my dream political match-up. RIP Mayor Ford. [↑](#footnote-ref-1)