Team # 6

Project #1, Task 3: Program Code Rubric

The following is our grading rubric we will use to grade the coding portion of Task #3. Total points: 50

Program Specification and Correctness (60%)

4	The program compiles and there are no runtime errors. The program always works correctly and meets all specifications. Both the IRV and open party listing work properly.
3 X 305	The program compiles and there are no runtime errors. The open party listing algorithm is fully implemented and works properly for all test cases. The IRV algorithm is fully coded and most test cases work. All other specifications have been fully implemented with no errors. (Vice versa can occur for open party listing and IRV) → errors in IRV (check github)
2	The program compiles and there may be some runtime errors. The open party listing voting algorithm is fully implemented and works properly for all test cases. The IRV algorithm is fully coded and does not work for nearly all test cases. All other specifications have been fully implemented with some errors. (Vice versa can occur for open party listing and IR
1	The program compiles. Specification details are violated and the program exhibits incorrect behavior for nearly all tasks.

Readability (25%)

4 X	Code is easy to read and follow. It should be stylistically designed to include consistent indentation; use of whitespace; all variables, methods/functions, and files have meaningful and professional names; code is well organized and is designed to be reuseable. You follow OOP principles of design and do not have huge blocks of code in a single method. You pass arguments when required instead of just keeping everything in one class and use the concepts of encapsulation, inheritance, and polymorphism (if needed).
3	Code is easy to read and follow. Minor issues with consistent indentation, whitespace usage, naming of variables, and general organization. You follow some OOP principles of design and do not have huge blocks of code in a single method. You pass arguments when required instead of just keeping everything in one class and use the concepts of encapsulation, inheritance, and polymorphism (if needed).
2	Code is not always easy to read and follow. Many issues with consistent indentation, whitespace usage, naming of variables, and general organization. You follow little to no OOP principles of design and have huge blocks of code in a single method. You rarely pass arguments and use hard coding instead of following good coding technique.
1	Code is not easy to read and follow. Many issues with consistent indentation, whitespace usage, naming of variables, and general organization. You follow no OOP principles of design and have huge blocks of code everywhere. You rarely pass arguments and use hard coding instead of following good coding technique.

Documentation of Code (not accounting for Javadocs or Doxygen – separately graded) (15%)

4 X	The code is well commented. We are expecting comments to be in the code itself explaining what is happening for complex code, loops, and decisions. Not overly commenting code with irrelevant comments that add no real content.
3	The code is sometimes well commented. We are expecting comments to be in the code itself explaining what is happening for complex code, loops, and decisions. (e.g. Do not comment a loop by saying it is a loop. Tell us what the loop is doing.) Add some irrelevant comments that add no meaningful content .
2	The code is not well commented. The comments do not communicating clearly what is happening in the code especially for complex code, loops, and decisions.
1	Few comments with no real information being provided to the reader.

Efficiency (5%)

[✓ 4]	Efficient code decisions (e.g. use of loops instead of copy and pasting lines over and over.)
3	Some poor choices in approaches.
2	Many coding approaches could have been programmed faster or easier.
1	Nearly all coding approaches could be optimized or programmed for less memory or speed.

Assignment Specifications (5%)

[✓ 4]	All files are in their proper located location on GitHub. You followed the instructions fully.
3	You missed putting 1 or 2 files in their proper location.
2	You did not setup the directory structure properly and only some files are in their correct location and many others are not.
1	Files are not stored where they should be on GitHub.

((_____* .60) + (___* .25) + (___* .15) + (___* .05) + (___* .05))/4 * 50 = 46.25

Unit Testing and System Testing Rubrics -Project #1, Task 3

Unit Testing Rubric

4	Full coverage f or all public and protected methods including obvious boundary conditions. Everything is fully documented and understandable logs included as part of logging documentation.
3	Nearly full coverage for all public and protected methods including obvious boundary conditions. Everything is fully documented and understandable logs included as part of logging documentation.
2	All public and protected methods have one test. Everything is fully documented and understandable logs included as part of logging documentation.
1	All public and protected have only one test. There is insufficient documentation for the logs.

3.5 /4 * 60 = ~~45~~ 52.5

This score will be added to the program portion's score

Boundary cases not added

===

System/Integration Testing

4	Full coverage testing for the entire program for both plurality and droop quota. Have multiple test files covering boundary scenarios. Everything is fully documented and understandable logs included as part of logging documentation.
3	Near full coverage testing for the entire program for both plurality and droop quota. Have some files covering boundary scenarios. Everything is fully documented and understandable logs included as part of logging documentation.
2	Little coverage of testing for the entire program for both plurality and droop quota. Have no files covering boundary scenarios. Everything is fully documented and understandable logs included as part of logging documentation.
1	Little to no testing. Documentation is insufficient.
0	Not completed

3.5 /4 * 40= ~~30~~ 35

Add together points for total: ~~75~~ 87.5