**Main (main.h and main.cc)**

**main.h**

This is a header file for main.cc, and it includes imports for utils, scanner, and scanline. It utilizes namespace std so that we don’t have to type “std.suchandsuch” for every call. It also #includes configuration.h, simulation.h and myrandom.h

**main.cc**

This is the main driver program. It sets up all the file input/output, and creates a new: Configuration config, Simulation simulation, and MyRandom random. It configures the RN seed using myrandom, and obtains the station count spread, election day length, mean and deviation voting time from “config\_filename”. It then reads the precinct data from “pct\_filename”, and runs the simulation.

**Myrandom(myrandom.h and myrandom.cc)**

**myrandom.h**

This is a header file for myrandom.cc; It includes imports for scanner and scanline, and utilizes namespace std;

**myrandom.cc**

This generates random numbers(Normal, Double, Int, etc); We are utilizing this functionality to generate a random number(s) based on a seed, obtained from config. There is a constant value kDefaultSeed that we utilize if no such seed is obtained from config.

**Configuration(configuration.h and configuration.cc)**

**configuration.h**

This is a header file for configuration.cc; Includes utils, scanner, scanline and myrandom. Uses namespace std. Has a few constants, kDefaultSeed, kDummyConfigInt, and kDummyConfigDouble. Has much of the data that we need to set up, including: seed, election\_day\_length\_hours\_, election\_day\_length\_seconds\_, time\_to\_vote\_mean\_seconds\_ and many other related variables to waiting times, service times, arrival rate, etc.

**configuration.cc**

This reads a config file (called in main.cc) and sets all the variables mentioned above to their respective values. Includes a ToString to convert all the variables to a formatted string for output.

**Simulation(simulation.h and simulation.cc)**

**simulation.h**

This is a headerfile for simulation.cc, includes utils, scanner, scaneline, and utilizes namespace std. Also #includes configuration and onepct.

**simulation.cc**

Creates a new precinct OnePct new\_pct, and reads data to create said precinct. Iterates over the precinct data, and gets the expected number of voters. If the expected does not fall within our expected min/max, iterate again. If expected does fall within parameters, run the simulation for this precinct. Also contains a ToString to convert Precinct data to a formatted string.

**OnePct(onepct.h and onepct.cc)**

**onepct.h**

This is the headerfile for onepct.cc, and includes utils, scanner, and scanline. Also includes configuration, myrandom, onevoter, and includes the containers cmath, map, set and vector. Contains dummy constant values kDummyDouble, kDummyInt, and kDummyString. Several private variables including: pct\_expected\_voters\_, pct\_expected\_per\_hour, set stations\_to\_histo\_, vector free\_stations\_, multimaps, and more.

**onepct.cc**

Has normal accessors for data, and a ReadData to initialize all of the variables. ComputeMeanAndDev, CreateVoters, and DoStatistics all do precisely what they are called. RunSimulationPct utilizes config and random to create mins and maxes, RunSimulationPct2 to actually simulate voting at stations, and a standard ToString as well as a ToStringVoterMap to return formatted strings.

**Onevoter(onevoter.h and onevoter.cc)**

**onevoter.h**

This is a header file for onevoter.cc and includes utils, namespace std, a constant kDummyVoterInt. Has multiple private variables including sequence, time\_arrival\_seconds\_, time\_done\_voting\_seconds, and more.

**onevoter.cc**

Has Accessors for times(arrival, waiting) as well as station number. Has a default and a parameterized constructor, and has an assignstation function to assign a voter to a station, calculate how long it takes them to vote (waiting, start voting, done voting) and assign that total time value to a voter. Has a ConvertTime function to convert all times to standard times, and a ToString to return a formatted string that represents a voter.