Team Belleville Data Design Document
April 10, 2017

All of the data, questions and outputs are generated via csv files. Using the Canadian crime and census data files, our program is able to read those files and parse the information by removing the commas in between each set of information. Our program first verifies that the necessary data exists and parses the data to create a smaller csv file that has only the most important points of information in order for the other files to read the data much faster. Questions are generated by user input. The user is prompted with either choosing whether their question wants all the crimes or a specific type of crime. From that point they can either type the specific crime they want to learn about or move on to the location they want to learn about. The user can either select to learn about all of Canada as a whole or they can enter a specific location. Basically the user can create questions by selecting from a list of specific options in order to generate a question. Based on the selections, a csv file is created with specific letters that the program can read. The question script will first ask the user how many crimes they want to consider, giving them two prompts. Every crime or only one or a few specific crimes. Once the user enters their desired crimes, the program will ask for a location to look for. The user can answer every location in Canada or only one or a few specific locations. After this part, the program will once again prompt the user with their desired year range. The user can either input every year, a specific year, a range of years or two different years. Once the years have been input, the user will have to answer how they want their answer formatted. Based on whole number of actual incidents, rate per 100000 people or a percentage change in rate. After the user answers this portion, the program finishes and will begin putting all of the information taken from the user into a csv file. The program reads the years part first. The letter A represents all years, so the program will assume every year, Canada, or crime is selected. The letter O represents only, so that would mean a specific date or location or crime is selected. The

answer is then generated using the the crime or census data csv files and the newly generated question csv file. The readFile.pl script is supposed to take in the csv files and compares information from both files in order to output the specific information from the file. Once the program reads everything from the csv files, the answer is outputted for the user to see. The design of the software was changed because it was required to read multiple csv files off each other. We split the work up into multiple different sections and that was how our files were integrated together. Our idea for the design of this program was thinking about how the user would input their question and how that question was going to be used in order to retrieve specific information about certain data. We also thought about how the user could potentially break the program by entering whatever they wanted, which was why we restricted them to being able to choose from a certain list in order to generate a question that we could actually answer. Unfortunately we were unable to accomplish our goals of finishing the census data and the graphing portion of the assignment. Which would have potentially changed the overall design of the program. All of the data the user asked for is output into another csv file in which the user can use for their own liking.