Langara College
Department of Computing Science & Information Systems
CPSC1160 – Algorithms and Data Structures I
Lab_09: Files and Overloaded operations
Dr. Bita Shadgar

1. Instructions

a. Writing your program with nice style is part of your evaluation. Indentation, documentation, modularization and selecting good names for your variables and constants are important.

b. In order to **not to lose mark about indentation**, make sure that find and replace all the tabs in your final program with 4 spaces, then save the code and submit it. This is due to different size of tab in different word processors (editors). It happens sometimes that your program lose right indentation when it is open in a new editor.

c. Read whole assignment first and make sure that you understand different parts of assignment and due dates. If you have any doubt or you are not clear about the assignment, you should ask in lab sessions or office hours. There is no grantee to get answer for your questions in this regard via email out of those times.

d. Create a folder named **Lab9**.

e. Inside folder **Lab9,** create a file for each problem.

Problem 1: [10 marks] Online quiz (Filename: **Quiz.pdf**)

Answer Multiple Choice Quiz at www.cs.armstrong.edu/liang/cpp3e/quiz.html for chapters 13 and 14. Then save your results as an image and paste them into a text file. Finally convert text file to pdf version named Quiz.pdf.

Problem 2: [10 marks] Baby name popularity ranking (BabyNames.cpp)

The popularity ranking of baby names from years 2010 to 2015 are downloaded from www.ssa.gov/oact/Babynames and stored in files named **Babynameranking2010.txt**, **Babynameranking2011.txt**, …These files are in Names folder (download and unzip Names.zip file).

Each line contains a ranking, a boy's name, number for the boy's name, a girl's name, and the number for the girl's name. For example, the first two lines after header line in the file **Babynameranking2010.txt** are as follows:

1	**Jacob**	**22,095**	**Isabella**	**22,890**
2	**Ethan**	**17,993**	**Sophia**	**20,623**

So, the boy's name **Jacob** and girl's name **Isabella** are ranked **#1** and the boy's name **Ethan** and girl's name **Sophia** are ranked **#2**. 22,095 boys are named **Jacob** and 22,890 girls are named **Isabella**.

a. **[5 marks]** Having text files for years 2010 – 2015, write a function named **getStat** that receives a year and calculates some statistics about each year. It returns a string representing year, number of male new born (or male population MP), percentage of male population (MPP), number of female new born (or female population FP), percentage of female population (FPP), and total new born (or total population) which are separated by coma in the returned string.
For example, running **getStat("2010")** should return a string like following:

2010, 8039304, 51.12%, 7688043, 48.88%, 15727347

It means in 2010, there are 8,039,304 male new born and 7,688,043 female new born, which comprise respectively 51.12% and 48.88% of whole new born population (15727347).

b. **[5 marks]** Write a function that applies **getStat** function to generate a file named **stat.txt**. The first line of **stat.txt** is a header line like:

Year, MalePopulation, MPP, FemalePopulation, FPP, TotalPopulation

Also it's next 6 lines are the result of calling **getStat()** for different years between 2010 to 2016.

Hint: In order to do not lose mark, please put **Names** directory in the same path as your **cpp** files and use the relative file name to access the files. For example, something like:

```
ifstream input("./Names/Babynameranking2010.txt")
```

Problem 3: [10 marks] Combin all Files into One File (filenames: **Name.h, Name.cpp, allNames.dat, BabyNames.cpp)**

Write a function named **combinFilesToBinary** that extracts all data for different years 2010-2015 and combines all data into a binary file named **allNames.dat**.

Hint: You first need to define a class named **Name** that includes **name, year, frequency** (#male or #female)**, gender** (M or F), and **rank** data fields with appropriate methods. You can complete the needed methods by going through the assignment step by step.
Then, read from files and create two Name objects for each line of data (one for male, one for female), and save them into the binary file named **allNames.dat**.

Problem 4: [15 marks] Sort Name Objects (filenames: **BabyNames.cpp)**

Write a function named **sort** that sorts objects in **allNames.dat** file. Then write the sorted list into a binary file named **sortedAllNames.dot**. To compare two objects of Name class you **must** override operators <**,** <= and ==. Also to write/read into/from stream you need to override << and >> operators in the Name class.
To compare two Name objects, you should consider the following order on fields of object:

1. Frequency

2. Name
3. Gender
4. Year

In other words, for two objects *o1* and *o2*, you should first compare *o1.frequency* and *o2.frequency*, and based on the result, decide which one is bigger. However if *o1.frequency* is equal to *o2.frequency*, you should use the next field i.e. **name** and compare *o1.name* to *o2.name*. In case that *o1.name* is equal to *o2.name*, you must compare the **gender** (assume **M>F**), and so on.

Problem 5: [5 marks] List 20 (Most/Least) Popular Names (filenames: **20Names.txt, BabyNames.cpp)**

Write a function named **find20Populars** that reads top 10 most popular and 10 least popular names from **sortedAllNames.dot** binary file and writes them into a text file named **20Names.txt**. The first line of **20Names.txt** is its header like:

Name, Gender, Frequency, Year, RankInYear

This line should be followed by information of 10 most popular name and 10 least popular name in next 20 lines. Information in each line should be separated by coma (similar to header).

Due date

- By the end of the lab time, demonstrate **Problem 2**.
- By 11:59pm on Monday 27, March 2017, submit a zip file named Lab9.zip which includes 4 files named **Name.h**, **Name.cpp**, **BabyNames.cpp,** and **Quiz.pdf** to D2L.