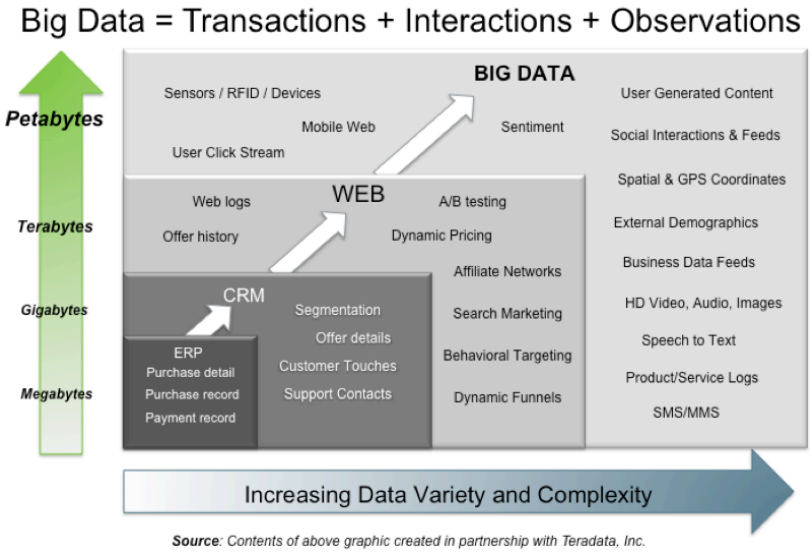
(http://blog.oxfordknight.com/)

CS101 Homework 5: Big Data

Prof Tejada
Program and report due:
11:59pm Monday, April 1
Design document due:
11:59pm Monday, March 25

With the advent of the internet, the Web, social media and smart phones, we all have become massive data producers, daily generating many bytes of email, texts, pictures, tweets, movies, documents, purchases, etc. According to IBM "Every day, we create 2.5 quintillion bytes of data — so much that 90% of the data in the world today has been created in the last two years alone." Volume, Velocity, and Variety are 3 metrics for computing at scale (aka Big Data). Volume measures the amount of data in bytes. Velocity measures the rate at which the data is generated. Variety captures the richness and complexity of the data. Check out this youtube video with a brief tutorial: http://www.youtube.com/watch?v=7D1CQ_LOizA&feature=youtu.be



For this assignment we will use a publicly available dataset provided by the Social Security Agency – the set of popular baby names from 1880-2011. Check out their website to interact with the dataset (http://www.socialsecurity.gov/babynames/).

Your program will use this dataset located here:
http://www-bcf.usc.edu/~stejada/csci101/HW/HW5/data/names.zip

In this zip file are all the baby names divided into text files by year. The baby names for year 1880 are in file "yob1880.txt". Write a program that can take as input on the command line a person's name, sex and year of birth then generate a graph visualizing the popularity of the name starting at 1880 til 2011. It should also find the most popular female and male names in the person's year of birth and generate a second graph visualizing how the person's name compares to them throughout the years.

This homework will have you:

1. Create your own C++ class and the syntax of defining data members and methods.
2. Use File I/O to read/write text files containing structured data
3. Use command line arguments
4. Create visualizations of your data using the myro library (lab 5)

The File Format: This assignment requires you to use File I/O techniques to read the baby names from files. All of the baby name files represent all information as text and use the format specified below. Each line will have the data for exactly one baby. The first attribute is a baby name separated by a comma, then the baby's sex (M or F) separated by a comma, then the number of babies born with that name. Each line has the format: *<babyname>,<M/F>,<frequency>*. Also for each file all the female names are listed first from the most frequent to the least frequent, then all the male names are listed. For example from file "yob2009.txt," the name Isabella is the most popular female name with 22,067 babies born in 2009 with that name. The file lists all the female names then starts with the most popular male name, which is Jacob with 20, 858 babies born with that name in 2009.

```
Isabella,F,22067
Emma,F,17716
Olivia,F,17246
Sophia,F,16743
Ava,F,15730
Emily,F,15204
.
.
.
Jacob,M,20858
Ethan,M,19664
Michael,M,18677
Alexander,M,18025
William,M,17696
Joshua,M,17418
Daniel,M,17336
Anthony,M,16139
```

If on the command line you run your program with these parameters:

```
./bigdata William M 2009
```

It should not only find the statistics for the name "William" from 1880-2011, but it should find the statistics for the most popular female name "Isabella" and the most popular male name "Jacob" as well.

Data Visualization

Visualization is another V word associated with Big Data. There are many ways to graphically represent data in meaningfully. Here are a few links for you to investigate to get some ideas on how you want to graphically visualize your data.

http://blog.twitter.com/2012/11/visualizing-2012-election.html
http://aworldoftweets.frogdesign.com/
http://www.densitydesign.org/2012/04/5351/

Not interested in baby names? Find Your Own Big Data!

It should be on something that interests you:

1. Think simple
2. Find a dataset with 3-5 columns.
3. Once you've got your dataset, do the following:
4. Write a couple of sentences about what your dataset contains (column names, types) and why you chose the dataset. Send me an email by Monday, March 25.
5. Your goal will be to teach me something about your dataset.
6. Make some graphs.

Here are some good sources for publicly available datasets:

Google Fusion Tables http://www.google.com/fusiontables/

Twitter API http://api.twitter.com

Facebook Open Graph API http://developers.facebook.com/docs/reference/api/

Geonames API information on any geographic location
http://www.geonames.org/export/

Yelp API http://www.yelp.com/developers/documentation

Flickr API http://www.flickr.com/services/api/

New York Times API http://developer.nytimes.com/

Kiva API http://build.kiva.org/docs/getting_started

Other data: http://www-bcf.usc.edu/~stejada/csci101/HW/HW5/data/

Design Document

Write down electronically or on paper the answer to the following questions **BEFORE** beginning. You MUST do this **BEFORE** you begin to program or write code.

- Write a description of your algorithm. What are the individual steps? What are your functions? What are the input parameters of each function?
- What data needs to be maintained for each baby? For all babies? For the visualization graph?
- What type of user input do you need? How will your program use this input to decide what statement to execute next? How will you validate this input
- What data needs to be maintained for the total execution of your program (i.e. across several graphs of baby names)? What type?
- What kind of loops should then be used?
- What is your development plan? In what order will you implement and test your program? What is your development schedule? When will each stage of your program be completed?
- How are you going to gather frequency of the baby name from all the files?
- How are you going to visualize these statistics in a graph?
- How are you going to add the statistics of the most popular baby names to the graph?
- If you are not going to use the baby dataset you must submit your data with your design document
- Blackboard submission (Due Monday March 25, 11:59pm)
    - Zip your Design Document and submit the zipped file to Blackboard.

Requirements

Your program shall meet the following requirements for features and approach:

1. You may **NOT** use global variables. You instead need to pass appropriate arguments either by value or by reference/pointer.
2. Allow for command line arguments for baby name, sex, birth year, name of output graph file
3. Menu display with instructions
4. Prompt the user to enter an action to perform.
5. Receive and validate input from the user
6. Perform the action (enter baby name, gather baby statistics, and create graph of baby statistics)
7. Display status or results of action.
8. Use a class to represent a baby with member name, sex, birth year, and array/vector for 130 name frequencies from 1880-2011
9. Create functions in the program code implementation.
10. Use the myro image processing library, used for Lab 5 Chroma key, to create two graphs of baby statistics

11. Graphically visualize the baby statistics for the given baby name over all years for the given baby's sex.
12. Graphically visualize the baby statistics for the top male and female baby names in the given year over all years together with the statistics for the given baby name.
13. Use file I/O to read files
14. Save graphs as jpg files
15. Allow the user to repeat the search and visualization for another baby name
16. Exit upon user choice where appropriate

Report

Include the following items in your report. **Be sure to have your name and USC ID on the first page at the top.**

1. Your answers from the design document (Due Monday March 25, 11:59pm) and also include a discussion of any design choices you made, any known errors, and any other information that will help us in grading your assignment.
2. A printout of 2 run-throughs of your program, one for each graph.
3. Place your 'big_data.cpp' file, 'baby.h' file, 'baby.cpp' , your dataset, and Report in a ZIPPED folder and submit that ZIP file. Failure to place the files in a ZIP file will lead to point deductions. Then submit the ZIP file via Blackboard.
4. Blackboard submission (Due Monday, April 1, 11:59pm)
    a. Zip your Report and your C++ program files together and submit the zipped file to Blackboard.

Please also note that you must submit your code using the USC Blackboard System since the Blackboard System timestamps your submission. You should also verify what you have submitted is what you intended to submit. Please note that **it is your responsibility** to ensure that *you have submitted valid submissions, meaning that **your code must run on the provided Ubuntu image**, which is the machine that it will be graded on*.

Late Policy

All homework must be turned in on time. Late submissions will receive severe penalties. If you submit within 24 hours after the grace period, you will receive 80% of your grade. If you submit within 48 hours after the grace period, you will receive 50% of your grade. If you are unable to complete a homework assignment due to illness or family emergency, please see the instructor as soon as possible to get an extension. A doctor's note is *required* as proof of illness or emergency. In general, when you get sick, it's best to see a doctor and get a note just in case you may need it later.

Req. / Guideline	Wt.	Score	10 (Excellent)	8 (Good)	5 (Poor)	2 (Deficient)	(0) Failure
Design Document and Final Report	1		Well-organized, detailed description of algorithm, reasonable development and testing plan and schedule, provides insight into your decisions		Unclear description of algorithm, unrealistic development and testing plan or schedule, providing little insight into your decisions		Not submitted or submitted late
Search for given baby name statistics over all years	2		Reads all Baby name fields correctly, creates baby object and displays in well formatted out	Reads all Baby name fields correctly, creates baby object and displays in an adequately formatted output	Either misses a Baby while reading from the file or does not display a Baby or a field. Display format is adequate.	Gross error in reading Babies or display their info.	Not implemented
Find top baby name statistics for given year	1		Reads all Baby name fields correctly, creates baby object and displays in well formatted output	Reads all Baby name fields correctly, creates baby object and displays in an adequately formatted output	Either misses a Baby while reading from the file or does not display a Baby or a field. Display format is adequate.	Gross error in reading Babies or display their info.	Not implemented
Visualization	3		Clear, detailed graphical visualizations of one baby object stats only and another with top baby statistics		Only one clear, detailed graphical visualizations of the given baby stats or given top baby statistics	Gross error in creating visualizations.	Not implemented
Menu display & Command line arguments	1		Menu prints correctly and can use command line arguments correctly	Menu prints slightly improperly	Menu prints incorrectly or uses command line arguments incorrectly	Command line arguments do not work.	Not implemented
C++ Class	1		Classes define correct methods and are used correctly (data is encapsulated correctly, baby objects created/allocated correctly, etc.)		Classes define some correct methods but are missing others (like constructors, etc.) OR classes are used inappropriately.	C++ classes are defined but not used.	Not implemented
Code Org.	1		Well-organized code (indented /readable) AND commented such that another person could easily follow what the code was attempting to accomplish as well as providing insight into your decisions	Acceptably organized code (indented /readable) AND commented in a reasonable manner to allow someone else to reasonably follow the code and provided some insight into your decisions	Poorly organized code (indented/ readable) OR comments were disorganized and provided little insight into the code and implementation decisions.	Poorly organized code (indented/ readable) AND comments were disorganized or provided little insight into the code and implementation decisions.	Completely unorganized code and lacked helpful comments.
Late			-20% (1 Day Late)	-50% (2 Days Late)	-5 (Submission Instructions)		
TOTAL							