SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

ENSC 251 D100 – Software Design and Analysis for Engineers (4 sem. hrs.)
Fall 2017

Lab 2

Assigned	the week of September 18, 2017
Due	Part A: before the end of the lab period. Part B: before Sat September 23, 2017 @ 9:00am.

This is an individual assignment.

- You may consult with professor and TA about any aspect of the assignment.
- You may consult with other students only in a general way, e.g., about debugging or C++ issues, or questions about wording on the assignment.
- You cannot actively work with another student in this assignment.

General Information

- **Lab**
  - We will be using the ESIL lab (ASB 10803) for all lab sessions.
  - Sessions:
    - LA01, Tu 2:30 PM - 4:50 PM, ASB 10803, Burnaby
    - LA03, Fr 9:30 AM - 11:50 AM, ASB 10803, Burnaby
    - LA04, Fr 2:30 PM - 4:50 PM, ASB 10803, Burnaby
- **Software Development Environment**
  - This semester we will be using CentOS Linux environment conjunction with GNU's gcc. As for text editor, there are many choices. For example, vim (http://www.vim.org), Code::Blocks (http://www.codeblocks.org/), and Eclipse. They all have cross platforms support (Windows, Linux, and Mac).
- Example codes will be provided to you. Also, in each of the files you submit, please include the following brief declaration on the top of the file.

```
// I declare that this assignment is my own work and that I have correctly acknowledged the
// work of others.  I acknowledged that I have read and followed the Academic Honesty and
// Integrity related policies as outlined in the syllabus.
//
// _____ (PRINT YOUR NAME HERE) ____      _____(DATE)______________
//
// ____ (STUDENT ID) ______
//
// (if this is group project, please list all the team members at the following space:
//
// _____ (PRINT YOUR NAME HERE) ____      _____(DATE)______________
//
// ____ (STUDENT ID) ______
```

1. Specifications

PART A

Follow the instructions in the git tutorial slides (https://evangeliney.github.io/slides/git_tutorial)

Step 1-4

- Fork the git_tutorial repository (https://github.com/SFU-2017-3-ensc251/git_tutorial) from our class organization on GitHub.
- Clone the forked repository to your lab computer using the terminal.

Step 5-10

- Modify the code by adding simple cout statements that will print your name. Then include some comments to identify your contribution.
- Compile the code using the makefile.

Step 11

- Push the modified code and the new executable back to your forked repository on GitHub.

Step 12

- Submit a pull request to the original git_tutorial repository before the deadline.
- Put your SFU ID (your SFU email without the @sfu.ca) in the title section.
- Put your name in the comments section.

You will be invited to our class organization SFU-2017-3-ensc251 on GitHub after the deadline.

PART B

Write a program that inputs two string variables, first and last, each of which the user should enter with his or her name. First, convert both strings to all lowercase. Your program should then create a new string that contains the full name in Pig Latin with the first letter capitalized for the first and last name. The rules to convert a word into Pig Latin are as follows:

- If the first letter is a consonant, move it to the end and add “ay” to the end.
- If the first letter is a vowel, add “way” to the end.

For example, if the user inputs “Erin” for the first name and “Jones” for the last name, then the program should create a new string with the text “Erinway Onesjay” and print it.

In your program, be sure to include code for proper error handling.

Use the example zip file from lab 1 as a starting point. Create *.cpp file as needed. Modify the makefile such that it will compile your code into a binaries executable.

a) Create a directory with your name, e.g. "\LastnameFirstname", where Lastname is student's last name and Firstname is the first name.
b) Save the files (*.cpp, other files, and makefile) in this directory. Uses these files as a starting point to write the following program.

Then Zip up the directory "\LastnameFirstname" and the files within this director into a zip file "2017-3-ENSC251-LastnameFirstname.zip." Submit the zip file to Canvas before the deadline.

Resources

- C++ Formatter https://codebeautify.org/cpp-formatter-beautifier
- Vim Basics - https://www.howtoforge.com/vim-basics
- Common Linux Commands http://www.dummies.com/computers/operating-systems/linux/common-linux-commands/

1. Rubric for marking

PART A (10%)

Marks	Task
1	Completed part A
0	Others

PART B (90%)

Criteria	Ratings				Pts
Program Specifications / Correctness	**Excellent** - No errors, program always works correctly and meets the specification(s). 50.0 pts	**Adequate** - Minor details of the program specification are violated, program functions incorrectly for some inputs. 40.0 pts	**Poor** - Significant details of the specification are violated, program often exhibits incorrect behavior. 30.0 pts	**Not met** - Program only functions correctly in very limited cases or not at all. 0.0 pts	50.0 pts
Readability	**Excellent** - No errors, code is clean, understandable, and well-organized. 20.0 pts	**Adequate** - Minor issues with consistent indentation, use of whitespace, variable naming, or general organization. 16.0 pts	**Poor** - At least one major issue with indentation, whitespace, variable names, or organization. 12.0 pts	**Not met** - Major problems with at three or four of the readability subcategories. 0.0 pts	20.0 pts
Documentation	**Excellent** - No errors, code is	**Adequate** - One or two places that could benefit from	**Poor** - File header missing, complicated lines or sections of	**Not met** - No file header or	20.0 pts

	well-commented. 20.0 pts	comments are missing them or the code is overly commented. 16.0 pts	code uncommented or lacking meaningful comments. 12.0 pts	comments present. 0.0 pts	
Code Efficiency	**Excellent** - No errors, code uses the best approach in every case. 5.0 pts	**Poor** - Code uses poorly-chosen approaches in at least one place. 3.0 pts	**Not met -** Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion 0.0 pts		5.0 pts
Assignment Specifications	No errors 5.0 pts	Minor details of the assignment specification are violated, such as files named incorrectly or extra instructions slightly misunderstood 3.0 pts	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood 0.0 pts		5.0 pts
Total Points: 100.0					

-END-