University of Technology, Sydney

Faculty of Engineering and Information Technology
School of Software

FACULTY OF INFORMATION TECHNOLOGY

31251 – DATA STRUCTURE AND ALGORITHMS
32510 – OBJECT ORIENTED PROGRAMMING IN C++
AUTUMN 2011

ASSIGNMENT 2 - DUE 11:59pm, Friday 6/5/2011

This assignment is worth 15% of the total marks for this subject.

Requirement

For this assignment, you will write a C++ program that converts English into “Euro-English”.

Background

The European Commission has just announced an agreement whereby English will be the official language of the EU rather than German, which was the other possibility. As part of the negotiations, Her Majesty's Government conceded that English spelling had some room for improvement and has accepted a 5 year phase-in plan that would be known as "EuroEnglish": --

In the first year, "s" will replace the soft "c". Sertainly, this will make the sivil sevants jump with joy. The hard "c" will be dropped in favor of the "k". This should klear up konfusion and keyboards kan have one less letter.

There will be growing publik enthusiasm in the sekond year, when the troublesome "ph" will be replaced with the "f". This will make words like "fotograf" 20% shorter.

In the 3rd year, publik akseptanse of the new spelling kan be expekted to reach the stage where more komplikated changes are possible. Governments will enkorage the removal of double letters, which have always ben a deterent to akurate speling. Also, al wil agre that the horible mes of the silent "e"'s in the language is disgraceful, and they should go away.

By the 4th yar, peopl wil be reseptiv to steps such as replasing "th" with "z" and "w" with "v". During ze fifz year, ze unesesary "o" kan be dropd from vords kontaning "ou" and similar changes vud of kors be aplid to ozer kombinations of leters.

After zis fifz yer, ve vil hav a reli sensibl riten styl. Zer vil be no mor trubls or difikultis and evrivun vil find it ezi tu understand ech ozer.

ZE DREM VIL FINALI KUM TRU!!

Program Action

1. Your program will get the name of the file containing the English to be converted as a command line argument.

2. It will then read the file and build up the English message, character by character, in a linked list. Each node in the list will contain a single character.
3. You will then apply the following rules to the message contained in the link list. These rules will convert the English to Euro-English.
   - Replace all ‘c’ with ‘s’ if followed by the characters ‘e’, ‘i’ or ‘y’, otherwise replace with ‘k’.
   - Replace all instances of ‘w’ with ‘v’.
   - Replace all instances of “ph” with the character ‘f’.
   - Replace all double characters with a single instance of the character.
   - If a word is more than 3 characters long and ends with an ‘e’ then it can be removed.
   - Replace all instances of “th” with ‘z’.
   - Replace all instances of “ou” with ‘u’.
   - Replace all instances of “ea” with ‘e’.
   - If a word ends with “ed” then replace it with ‘d’.

   Make sure with all of these changes that you keep the correct case.
4. As a final step, the program should then correctly free up all the memory used in the linked list.

For example, the output of the program might would look like

```
Here is a message from an important visitor:
"Greetings, earthlings, I have come to rule you!"
Hello, Mother, I can't talk right now,
I am being harangued by a little green thingy.

Her is a mesag from an important visitor:
"Gretings, erzlings, I hav kom to rul yu!"
Helo, Mozer, I kan't talk right nov,
I am being harangud by a litl gren zingy.
```

Details

While there are other data structures that would work, you must implement a “linked list chars” using the template list from the STL <list>. Do not use any other data structure to store the text . See assessment section.

To help you understand how to use the STL list object I have included a file `exstdlist.cpp` in the assignment 2 sections of `\pub\prprog` and the DSA web site. Have a look at this code, compile and run it to see how it works.

A ‘word’ is defined as a string of alphanumeric characters that is preceded by a space or punctuation character or no character and followed by a space or punctuation character or no character.

Benchmark Program

To clarify how the program works, a benchmark executable version has been placed on the server. You can run it by typing the following command

```
/home/glingard/euro message
```

Where `message` is the name of a file containing English for conversion. To give you an idea of the size of this assignment, the source code for the benchmark program, including whitespace and comments, takes about 191 lines of code.

Assignment Objectives

The purpose of this assignment is to demonstrate competence in the following skills.

- ❑ Program design
- ❑ Command line arguments
- ❑ Handling template code and the STL
- ❑ Handling files
- ❑ Manipulation of linked lists

These tasks reflect all the subject objectives apart from objective 4.

As part of your subject workload assessment, it is estimated this assignment will take 14 hours to complete.

Queries

If you have a question, please contact the subject coordinator as soon as possible.

Gordon Lingard
glingard@it.uts.edu.au
9514-7935
Room 04.559, Building 10

However, for frequently asked questions a FAQ file will be put up. Please check this file before emailing the coordinator with a question.

```
/pub/prprog/assign/assign2/faq.txt
```

There is already an assignment FAQ in the DS & PP web site. You should read this.

If serious problems are discovered the class will be informed and an update will be included in the following file

```
/pub/prprog/assign/assign2/errata.txt
```

Please keep a look out for this file.

PLEASE NOTE. If the answer to your questions can be found directly in any of the following.

- ❑ subject outline
- ❑ assignment specification
- ❑ faq.txt
- ❑ errata.txt
- ❑ UTS Online discussion board
- ❑ DS & A web page

You will be directed to these locations rather than given a direct answer.

PLEASE NOTE. Please do not send email in HTML format or with attachments. They will not be read or opened. Only emails sent in plain text format will be read. Emails without subject lines will be automatically deleted by the junk mail filters I have in place.

Assignment Submission

You will submit your program via the `submitass2` program. It is assumed that your C++ file is called `assign2.cpp` (although you may call it something else).

You must start in a directory that is within your home directory on `charlie` and then run

```
      /home/glingard/submitass2 assign2.cpp
```

This assumes that `assign2.cpp` is in the current directory.

The `submitass2` program will then perform a number of tests. These include.

Compiling
Your program <u>must</u> compile without fatal errors <u>or</u> warnings – using the `g++` compiler. Your program will be compiled with the following commands and options

```
      g++ -pedantic-errors -Wall -Werror
```

You can learn what these compiler options mean by running `man g++`.

PLEASE NOTE. Your program must compile on the student UNIX server. Programs written on Window's machines sometimes don't compile or run properly on the student server.

Style Feedback Program
Your program will be run through a style feedback program, which will check that your code meets a minimum style layout. If your program does not meet the standard a warning message will be printed. The style feedback program will display messages showing which lines of code do not conform to the standard and why they don't.

Code Feedback Program
Your program will be run through a code feedback program, which will check that your code meets minimum coding practices. If your program does not meet the standard a warning message will be printed. The code feedback program will display messages showing which lines of code are not acceptable and why they aren't.

Test Filter
Your program will be tested with 5 different sets of tests. The results of your program will be compared to the results generated using the benchmark `euro` program.

PLEASE NOTE. Make sure the output from your program is <u>***EXACTLY***</u> the same as that from the benchmark executable. <u>***ANY***</u> deviation of the output from your program to that of the benchmark executable will cause the test to fail.

Plagiarism Agreement
You will be required to agree to a statement that the assignment is your own work and that you haven't given your code to anyone else.

If all goes well, `submitass2` will reply with a message saying you have successfully submitted the assignment. It will also place in your current directory a file called `receipt.txt`.

You may submit your assignment as many times as you like. The last assignment received will be the one marked.

PLEASE NOTE. Only assignments submitted via the `submitass2` program will be accepted for marking.

receipt.txt

`receipt.txt` is your proof that you have submitted your assignment. Once you have received it, copy it to somewhere safe such as your home directory. Additionally, copy your C++ file into the same location. Do not modify them in any way. If you wish to continue developing your program then do it on a duplicate file.

The `receipt.txt` file contains three pieces of information

1. A line saying you have submitted your file and when you did it.
2. A checksum of your C++ file
3. A checksum of your receipt

If you tamper with your C++ file or the receipt then the checksums will be become invalid and therefore your receipt will become invalid. No actions will be taken if receipts have invalid checksums.

Acceptable Practice vs Academic Malpractice

- ❑ Students should be aware that there is no group work within this subject. All work must be individual. However, it is considered acceptable practice to adapt code examples found in the lecture notes, labs and the textbook for the assignment. Code adapted from any other source, particularly the Internet and other student assignments, will be considered academic malpractice. The point of the assignment is to demonstrate your understanding of the subject material covered. It's not about being able to find solutions on the Internet. You should also note that assignment submissions will be checked using software that detects similarities between students programs.
- ❑ Do not let anyone submit their assignment from your account. The `submitass2` program copies your assignment into a secure directory based upon your user login name. Anyone else using your account will have their assignment placed in your directory. Students who do this will find themselves reported to the Faculty for possible academic malpractice and misuse of Faculty resources.
- ❑ Participants are reminded of the principles laid down in the "Statement of Good Practice and Ethics in Informal Assessment" in the Faculty Handbook. Assignments in this subject should be your own original work. Any collaboration with another participant should be limited to those matters described in the "Acceptable Behaviour" section. Any infringement by a participant will be considered a breach of discipline and will be dealt with in accordance with the Rules and By-Laws the University. The Faculty penalty for proven misconduct of this nature is zero marks for the subject. For more information, see *http://wiki.it.uts.edu.au/start/Academic_Integrity*

Assignment Security

It is important to note that you have a responsibility to maintain the security of your assignment files. You can read more details about this in the Academic Malpractice section of the DS & A web site - *http://learn.it.uts.edu.au/dsa/*

Assessment

Marks will be awarded out of 20 based upon the following scheme.

- ❑ Between 0 and 8 marks will be awarded by the computer based on the number of tests passed. 2 marks each for the first 3 tests passed and 1 mark each for the final 2 tests passed.

PLEASE NOTE. Some students have been known to write their code to artificially pass the tests rather than solve the assignment problem. In such cases a reduced mark will be given for the tests, including being given a 0.

The following marks will only be awarded if a score of 5 or more is awarded for the tests and both the code feedback and the style feedback programs are passed without generating warnings.

- ❑ Between 1 and 9 marks will be awarded on the quality of your code design and the algorithms used. This will also include looking at the proper use on `const`, etc.
- ❑ Between 1 and 3 marks will be awarded on the presentation style of the program. This involves meaningful variable names, intelligent use of comments and so forth.

PLEASE NOTE. It is a fundamental requirement of this assignment that you use the STL linked list of characters to store the English, as outlined in the details section. Furthermore, the conversion of the English must be done on the linked list. Do not copy parts of the linked list to an array for the assignment. Students who use any other data structure, including arrays, will receive 0 for the assignment.

For further details of the marking scheme please check out the assignments section of the DS & A web page. *http://learn.it.uts.edu.au/dsa/*

Late Assignments, Extensions and Special Consideration

Please read the subject outline regarding late assignments and extensions. If you did not get the outline a copy can be found at the DS & A web site

Assignments that are late by less than one week will incur a penalty of 1 mark for each day or part thereof late. Assignments more than one week late will not be accepted under any circumstance as the assignment solution will have been made available by then.

An extension of one week will only be granted if there is a fully documented reason which merits it. The documentation must be presented to the Subject Coordinator before the assignment due date. Extensions longer than a week will not be granted under any circumstance. If a one week extension is granted that means the assignment will be accepted up to a week after the due date without penalty. It will not be accepted later than that.

Students may apply for special consideration if they consider that illness or misadventure has adversely affected their performance in the assignment. For more information go to

http://www.sau.uts.edu.au/exams_ass/spec_cons.html

Return of Assessed Assignments

The code of your assignment will be printed out and marked. It will list marking codes for particular types of programming issues. These codes can be found in the DS & A web site in the `assessment->assignment->assessment` menu section and will give a complete break down of your marks.

You will be notified by announcement on UTS Online where marked assignments can be picked up. The estimated return date is 20/05/11.

Getting Marks

You can check on your marks by running the following program.

```
/home/glingard/getmarks
```

Apart from your marks this program will give the following information.

- ❑ Given out – the date the assignment is given out.
- ❑ Due date – the date the assignment is due.
- ❑ Late date – assignments will be accepted up to one week after the due date but with a penalty of –1 mark per day or part thereof late.
- ❑ Marks released – The date the marks are released.
- ❑ Close date – The assignment is closed and the solution will be released. The close date will be one week after the assignments are marked and given back.

PLEASE NOTE. It is your responsibility to check that I have received your assignment and given you a mark. Even if you have a receipt you should check your mark and inform me if there is any problem before the close date. The `getmarks` program allows you to do this very easily. Unless you have a valid receipt or there are exceptional circumstances (e.g. serious medical conditions) no further correspondence regarding the assignment will be entered into after the close date.