CS 341
Programming Languages
Spring 2021

Project #01

Complete By:	Noon on Wednesday, Feb 10th
Policy:	Individual work only, no late submissions
Assignment:	Create your own container and iterator
Submission:	submit electronically on Gradescope

Reading

In "A Tour of C++", read the following chapters (skimming them may help if you're still feeling uncertain after our discussion of iterators on 2/1):

- Chapter 6: library overview
- Chapter 9: containers
- Chapter 10: Algorithms

Exercise: Building an iterator

A template is provided of a character container (string) and an iterator which traverses the container one character at a time. You must adjust this template as follows:

1. Create a new iterator for the class which traverses the container one word at a time. Find a word according to the following algorithm: The first letter in the string (a-z or A-Z) is the first item output by the iterator. Each following word begins with a letter (a-z or A-Z) after a space, ", or (character ("(). This is a forward input iterator, meaning only the constructor, *, ++, != and == operators must be defined. The * operator should return a capitalized form of the letter at that location. The class definition of this iterator (acronymIterator) is provided, though not populated with the necessary methods.
2. Adjust the characterParser class to use the acronymIterator instead of the charIterator. You can test whether this is working because the loop in main will output the first letter of each word, forming an acronym instead.
3. Fix the printAllCharacters function to output all of the characters in the string, using the original charIterator.

Here are some sample inputs and outputs

Input:
a
abc
hello world
A cool reading object never yet made
"Words" (phrases) may "contain" 3 or more Other :characters.

Output as an acronym:
A
A
HW
ACRONYM
WPMCOMO

Output with this input and the given main.cpp file:

```
a
Input is : a
One of the words begins with an a or A
There are 1 vowels in the acronym
Input as Acronym : A

abc
Input is : abc
One of the words begins with an a or A
There are 1 vowels in the acronym
Input as Acronym : A

hello world
Input is : hello world
None of the words begin with an a or A
There are 0 vowels in the acronym
Input as Acronym : HW

A cool reading object never yet made
Input is : A cool reading object never yet made
One of the words begins with an a or A
There are 2 vowels in the acronym
Input as Acronym : ACRONYM

"Words" (phrases)  may "contain" 3 or more Other :characters.
Input is : "Words" (phrases)  may "contain" 3 or more Other :characters.
None of the words begin with an a or A
There are 2 vowels in the acronym
Input as Acronym : WPMCOMO
```

The code is provided over 7 files.
main.cpp – contains the code to run the program, take input and produce output
no changes need to be made, the grader uses its own main file to test your characterParser
characterParser.h – contains the definition of the characterParser class, including the forward declaration of the iterator classes for this container of characters
change this class to use acronymIterator instead of charIterator
characterParser.cpp – contains the implementation of the functions defined in characterParser.h
continue changing the class to use acronymIterator instead of charIterator,
modify printAllCharacters to continue to use charIterator.
charIterator.h – contains the definition of the charIterator class, nested within the characterParser
no changes to make, this iterator is provided as an example of how to implement an iterator, specifically a forward input/output iterator reading one character at a time
charIterator.cpp – contains the implementation of the charIterator class
no changes to make, provided as an example of a forward iterator impelmentation
acronymIterator.h – contains the definition of the acronymIterator class
mostly empty, up to you to define based on the description in 1. above
acronymIterator.cpp – contains the implementation of the acronymIterator class
mostly empty, up to you to implement based on the description in 1. above.
You can remove the default constructor provided as an example
Note that this is a forward input iterator, not a forward input/output iterator

Electronic Submission

For this project, you must submit 4 files, the two acronymIterator files (.h and .cpp), and the two characterParser files (.h and .cpp). Each of these files should include your name in the header comment at the top of the file. When you are ready to submit, login to Blackboard, find Assessments in the left hand side, then follow the link to Gradescope (or navigate to Gradescope directly), and then submit your files to "Project 1". You may submit as many times as you want, but we grade only the submission you indicate (generally the last submission, let us know via piazza if you want an earlier submission graded and don't know how to do so).

Policy

Late submissions not accepted. All work is to be done individually — group work is not allowed. You may discuss the provided code and how it operates on Piazza. You can not publicly post your code for others to see. While we encourage you to talk to your peers and learn from them, this interaction must be superficial with regards to all work submitted for grading. This means you *cannot* work in teams, you cannot work side-by-side, you cannot submit someone else's work (partial or complete) as your own. The University's policy is available here:

https://dos.uic.edu/conductforstudents.shtml .

In particular, note that you are guilty of academic dishonesty if you extend or receive any kind of unauthorized assistance. Absolutely no transfer of program code between students is permitted (paper or electronic), and you may not solicit code from family, friends, or online forums. Other examples of academic dishonesty include emailing your program to another student, copying-pasting code from the internet, working in a group on a homework assignment, and allowing a tutor, TA, or another individual to write an answer for you. Academic dishonesty is unacceptable, and penalties range from failure to expulsion from the university; cases are handled via the official student conduct process described at https://dos.uic.edu/conductforstudents.shtml .