Assignment 1
unfay ithway igpay atinlay
EE422C - The University of Texas at Austin

You will be working individually, no collaborating.
Points: 20 points total (2 for the analysis, 5 for the design, and 13 for the program).

Purpose – A compiler is a language translator, and so is the solution to this problem.
Pig Latin is a twist of English for people who don't want others to know what they're communicating about. Your task is to write a program that translates English phrases into their Pig Latin equivalent.

Schedule:
On this assignment you are required to adhere to the following schedule of events. Failure to do so will result in point deductions from your score.

1. **Read, parse, analyze** and attempt to fully understand the problem statement. Delineate any assumptions you are making and any questions you have about the problem you are to solve.
2. Submit a PDF version of your analysis before midnight on the designated due date. No submitted list of questions or assumptions will be taken to mean that you did not analyze the problem (i.e. 0).
3. Then you are to **design a solution** (**algorithm + functional block diagram**) on paper that solves the given problem. This design is the PLAN for your high-level program logic. If you have difficulties with doing this then go see one of the TAs during his lab hours. You should have learned basic program design techniques in EE312. If you are clueless about how to start doing this then see one of the TAs ASAP. Don't delay!
4. Submit a PDF version of your design before midnight on the designated due date. A ***workable*** design is worth 5 of the 20 points. If you miss this deadline you get a 0 on the design phase.
5. **Complete and submit your Java program** (.java file) using the Blackboard assignment manager before midnight on designated due date. A fully correct program is worth 13 points, 2 of which will be for conformance with the given Java coding standards (on the BB). A late program is worth 0 points.

Tips on accepting help from me or the TA. If you come to us for help we will first ask to see your current version of the design – show it to us no matter how rough it is. If we tell you that it or some part of it won't work, then make us explain why it won't to your satisfaction. We will try to help you formulate your solution; whatever that might be!! Remember that there are many different designs that can solve this problem – not just one – **and one that works is all that is required**.

Assignment Requirements

Design**,** write and run a Java program that will convert a sequence of given English language phrases into Pig Latin, according to the translation rules of Pig Latin given below.

A. Input: Using the given Translator class (supplied on the BB), the input to the program will be one English language phrase at a time to be translated into pig latin. The input lines for translation will come from a file, whose filename is to be specified in the command line argument. The translator class will handle reading the file, one line at a time. The file may contain multiple lines, but the template code we provide handles this.
B. Output: for each input phrase you are to output (to the screen) the original phrase and the resultant Pig Latin equivalent phrase, with appropriate identifying labels on each (see below for example).
C. **The given Translator class gives you a skeleton program to use. What you need to do is to fill in the *translate()* method to implement your solution.** Put in your code where it says so in the comments. You may add anything you need to the given code, but do not delete anything from it.

D. A sample set of test phrases to be processed by your program will be has been provided. You can create your own test phrases too.
E. **Your solution must use the Java String class and the associated String methods to manipulate the phrases. The phrases and various parts must be represented as Strings.**
F. Test your program well to make sure it works correctly. Use the given input data set when it becomes available. Remember that your program will be graded on the ENS LRC lab configuration **(Windows, Eclipse, with Java 6 mode set)** and must run successfully there.

Sample output: An example output from your program for a single input phrase might look something like this on the screen:

```
The phrase that was input is:
  Fun with pig latin!
The pig latin equivalent phrase is:
  unFay ithway igpay atinlay!
```

Pig Latin Translation Rules for English *Words*:

Here are the basic rules for translating English words into Pig Latin:

Rule 1. For words that begin with a single consonant, take the consonant off the front of the word and add it to the end of the word. Then add ay after the consonant. Here are some examples:
cat => atcay
dog => ogday
simply => implysay
noise => oisenay

Rule 2. For words that begin with double or multiple consonants take the group of consonants off the front of the word and add them to the end, adding ay at the very end of the word. Here are some examples:
scratch => atchscray
thick => ickthay
flight => ightflay
grime => imegray

Rule 1 and Rule 2 cannot be applied simultaneously.

Rule 3. For words that begin with a vowel, just add yay at the end. The letter y is a consonant when found at the beginning of a word. For examples:
is => isyay
apple => appleyay
under => underyay
octopus => octopusyay

Rule 4. Hyphenated words are treated separately with "-" placed in between. For example:
well-documented => ellway-ocumentedday

Rule 5. Contracted words are treated as one word. For example:
y'all => 'allyay
don't => on'tday

Rule 6. All other words and symbols not covered by rules 1-5 are not to be changed. Rule 6 is applied to non-words such as H1N1 or H*&N (i.e. not punctuation marks). It also applies to non-alphabetic words, or words with non-alphabetic characters embedded in them. When punctuation is included inside of a word, you can do whatever you want, except crashing. Punctuation symbols are defined as:, " (...) : ; ' . ! ?
For examples:
: => :
897 => 897

Additional tips and considerations:

- Coding standards - Starting with this assignment you must follow the class coding standards (see them of the class blackboard page). These standards ensure your program is readable by others. Failure to follow these standards shall result in deductions on your assignment.
- We may be using your program again in future assignments, so design it for further adaptations and document it well. Also comment your program so that its logic could be explained to your Grandmother (for example).
- Copying all or part of someone else's program is considered plagiarism, which will be dealt with according to the policy in the course syllabus.
- When in doubt, ask a TA for help, as opposed to assuming stuff