**Project 4 | State of the Union Speeches**

The State of the Union speeches of U.S. Presidents are available online. Here are some examples –

* <https://www.presidency.ucsb.edu/documents/annual-message-the-congress-the-state-the-union-3>
* <https://www.presidency.ucsb.edu/documents/app-categories/spoken-addresses-and-remarks/presidential/state-the-union-addresses>
* You can download and save each speech as a text file

Design, implement and test a Python application that will –

* Allow user to process multiple State of the Union speeches
* Analyze and display the statistical information of the select speeches
* Generate one **Word Cloud** for each speech
* Discuss the **readability (scores)** of each speech

**Academic Honesty Policy Reminder** | **Do your own work** – each submitted project will be compared against other submissions from current and previous semesters

**Requirements –**

* At a minimum, your application should –
  + Process **one or more (multiple)** speeches; each speech will be provided as an input (.txt) file
    - Prompt user for the filenames if you like
  + Display the following statistical information of each speech –
    - total word count
    - average word length
    - average sentence length
    - 15 of the most frequently used words (sorted by occurrences in descending order), excluding a, an, the, and, etc.
    - 10 of the longest words, in ascending order
  + **Generate Word Clouds** showing the visualization(s) of word frequencies of the speeches (complete below tutorial first)
    - <https://www.datacamp.com/community/tutorials/wordcloud-python>
  + **Determine the readability of each speech**
    - <https://pypi.org/project/readability/>
    - **Discuss your finings; what does the “score” mean? (Don’t overlook this)**
  + **Include one additional enhancement of your choice**
    - Be sure to talk about it in your write-up
* **Other Expectations**
  + Each student is encouraged to decide (on their own) how best to handle compound words, numbers (10 vs. ten), possessive words (dog’s), etc.
  + Feel free to display the statistics per speech or per run; you are here to **present your findings,** as best as you can
* **Python Expectations**
  + Utilize functions, collections, Python libraries, and file processing techniques

**Assumptions**

* Document any assumptions in your write-up

**Write-up & Submission Requirements**

Review the provided rubric and understand project expectations, including documentation, CMSC206, and programming requirements.

Two artifacts MUST BE submitted for this project – Python code and a write-up.

At a minimum, the write-up needs to address -

* Approach, design & algorithm
  + **DO NOT** start coding your project immediately! Come up with a high level design of the project first
    - What’s your game plan to complete the project?
    - Break the project into smallest modules where applicable
  + **Complete this step first, then write your code**
  + Each student is welcome to expand on the design, if it makes sense. Students will not be penalized for going “above and beyond” the specifications of the project
* Test plan & test cases
  + What test cases did you run?
  + Capture test runs as screenshots in your write-up, as you run your test cases
  + I want to understanding your “thinking,” as to how you are testing your program
  + Each submission should be rock solid, with “zero bugs”
  + **I am a typical user and will be able to break your code with ease**
* Any assumptions that you are making for this project
* Highlight your learning experience and lessons learned
  + **I am very interested to learn about what you have done, how you did, etc.**
* Anything else that you want to share with me

Each student must submit one compressed (.zip) file back to the Assignment (link) with the following deliverables:

* Source code
* Write-up (in Word or PDF – one write-up per project)
* Name the compressed file as <lastname>\_project\_x
  + where x is the project number and your last name (e.g., Thai\_Project\_1.zip)
* Review provided instructions on how to submit the project carefully (don’t assume anything)
* **Double check your submission. I can only grade what’s being submitted**
* **I MUST BE able to compile, run and test every submitted project on my computer using a set of test cases.** Just as important, I READ your write-up first

**Not clear? That’s okay, but do ask your questions. “I did not know” or “I did not understand” is not good enough.**

Start working on each project immediately so that we can discuss any concerns or questions you have!

**Sample Test Runs (Your requirements are different!)**

![A screenshot of a cell phone

Description automatically generated]()

![A close up of a logo

Description automatically generated]()

![A close up of text on a black background

Description automatically generated]()

![A screenshot of a cell phone

Description automatically generated]()