PROGRAM DESCRIPTION

This program (Abbreviation Generator Program) is designed to read a list of names from a file, generate three-letter abbreviations for each name, and then choose the best abbreviation for each name based on a set of rules. This program follows a series of steps taking into consideration factors such as letter positions, word boundaries, and assigned scores.

The primary goal of the program is to create unique three-letter abbreviations for given names, considering specific rules and scoring mechanisms for each abbreviation. The rules include ignoring apostrophes, splitting names into separate words, and calculating scores based on letter positions within words.

Key Steps:

- Loading Letter Values: The program starts by loading letter values from a file (values.txt). This file contains assigned scores for each letter based on its position and frequency in the English language.
- Cleaning and Formatting Names: Names are cleaned and formatted according to the rules specified in the problem. This involves removing apostrophes, replacing non-letter characters, and ensuring CamelCase formatting.
- Calculating Scores: Scores are calculated for each letter in a name based on its position within a word. Special consideration is given to the first and last letters, and a bonus is added based on the letter's frequency in English.
- Generating Abbreviations: Three-letter abbreviations are generated for each name by considering all possible combinations of the first letter and two subsequent letters. The score for each abbreviation is calculated based on the previously determined scores for individual letters.
- Handling Duplicates: The program identifies and removes duplicate abbreviations that can be formed from different names. This ensures that an abbreviation is unique across the entire list of names.
- Choosing Best Abbreviations: For each name, the program selects the abbreviation with the lowest score. In cases where multiple abbreviations have the same score, all such abbreviations are retained.
- Writing Results to File: The final step involves writing the original names and their chosen abbreviations to an output file. The output file is named according to the specified convention.

Implementation:

The program is implemented in Python and organized into functions to enhance modularity and readability.

Program Testing:

Test data:
Cold

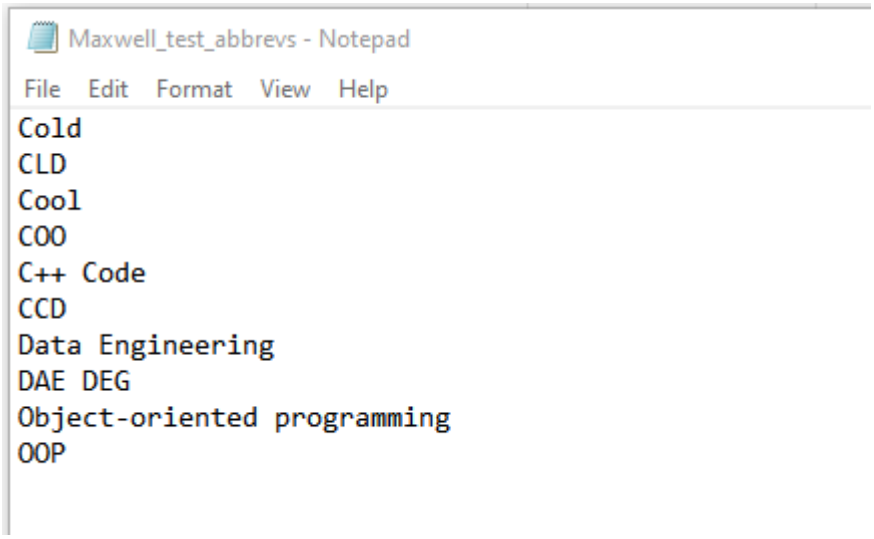
Cool

C++ Code

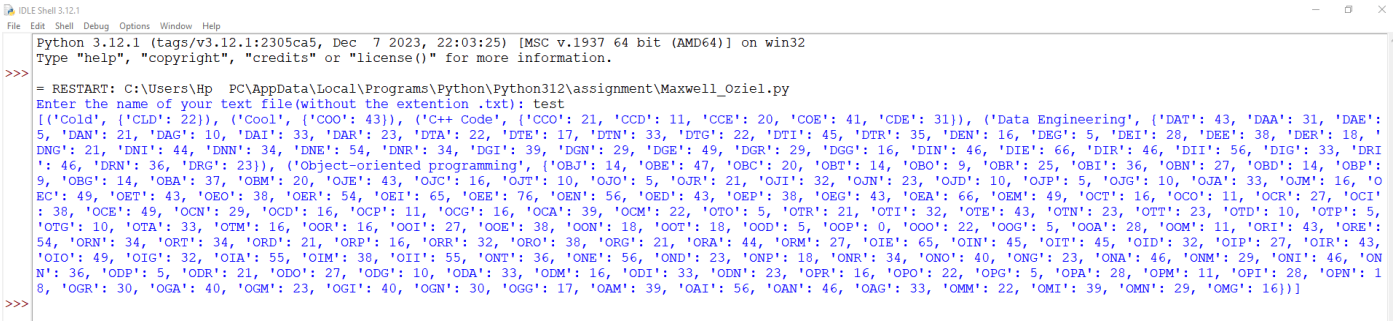
Data Engineering

Object-oriented programming

Test Result:



Created text file with the original name and it's chosen abbreviation



Result showing all possible abbreviations