**Technical Design Document For Chapter 2 Exercise**

**Name:** Jackson Hodge

**Date Created:** 1/29/25

**Program Description:**

This program asks the user to enter an email message. It will then scan the message to see the likelihood for it to be spam. The program will output a "spam score" and found spam words.

**Functions used in the Program (list in order as they are called):**

1. **Function Name:** spam\_dictionary

**Description:** This function creates a dictionary of 30 words/phrases that are commonly used in spam emails. If the email message the user enters contains words that are in the dictionary, it will contribute to the spam evaluation.

**Parameters:** This function does not take in any parameters.

**Variables:** spam\_dict is equal to the dictionary of spam words.

**Logical Steps:**

1. Set spam\_dict equal to the dictionary of spam words.

**Returns:** Return spam\_dict to be used for finding spam words in the user’s email message.

2. **Function Name:** calc\_spam\_score

**Description:** This function takes in the user’s input and finds spam words within it. For every spam word, the spam “score” has a value 1 added to it. Additionally, a list called spam words is created; for every spam word found, it is added to the list for output.

**Parameters:** This function takes in user\_email and spam\_dict.

**Variables:**

1. spam\_score
2. spam\_words

**Logical Steps:**

1. Initialize spam\_score and set it equal to zero.
2. Initialize spam\_words and set it equal to an empty list.
3. Start a for loop. For every word in the dictionary, if a word from the dictionary is in the user’s message (user\_email.lower(): make input all lowercase to make it not case sensitive) add 1 to spam score, and add the spam word to the spam\_words list.
4. Return spam\_score and spam\_words to be used for output.

**Returns:** spam\_score and spam\_words.

1. **Function name:** main

**Description:** This function gets input from the user and calls the other two functions to retrieve data for output. This function also determines the likelihood of the message being spam and displays the output.

**Parameters:** This function does not take in any parameters.

**Variables:**

1. spam\_dict
2. user\_email
3. spam\_score
4. spam\_possibility

**Logical Steps:**

1. Call spam\_dictionary by setting it equal to spam\_dict.
2. Get input for the email message through user\_email.
3. Call calc\_spam\_score by setting it equal to spam\_score and spam\_words.
4. Now main has retrieved all the data needed for the output.
5. Determine the likelihood of the message being spam with an if statement.
6. Print the output.

**Logical Steps:**

1. main is called.
2. spam\_dictionary is called inside main.
3. calc\_spam\_score is called inside main.

**Link to your repository:** <https://github.com/Jackson112607/COP2373-ProgrammingConceptsII>