**Technical Design Document Template**

**Name:** Jace Walker

**Date Created:** 2/3/2025

**Program Description:**

This program asks for the user to enter a message. This message is then checked for possible spam keywords against a list of typical spam words. A 'spam score' is calculated by counting the number of spam words used. At the end, the program will print out the spam score, the likelihood this is a spam message, and the spam words that were found.

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

1. **Function Name:** spam\_checker(email, spam\_list)

**Description:** This function takes an input and verifies if that string is contained in a list. If it is found, then that string is added to a new list.

**Parameters:** email- this is the user’s input

spam\_list- this is the list of typical spam words I made up

**Variables:** None

**Logical Steps:**

1. User’s one long string was broken up into many ‘substrings,’ so for every substring, python checks if it’s contained in common spam word list
2. If the substring **IS** contained in spam list, that substring is added to new list containing the spam words found in the message

**Returns:** Returns nothing, since the new list of found spam words was globally defined.

2. **Function Name:** spam\_score(email\_spam\_words)

**Description:** Finds length of new spam word list and returns a value depending on the length.

**Parameters:** email\_spam\_words- list of spam words detected in user’s input

**Variables:** total-used to hold value of length of list

**Logical Steps:**

1. Assigns length of new detected spam word list to a variable named ‘total’
2. If statements check the length of list and returns string

**Returns:** Returns likelihood of scam as a string

3. **Function Name:** main()

**Description:** This function handles the general print statements and calls the two previous functions

**Parameters:** None

**Variables:** user\_message- this is the user’s input email message

**Logical Steps:**

1. Print title of program and ask user for input
2. Input is lowercased and split to turn one long string into substrings
3. Calls spam\_checker function
4. Prints spam score using length of list (this gives the numerical score)
5. Runs and prints spam\_score function (this gives the string to display to user)

**Returns:** None

**Logical Steps:**

1. Calls main function
2. spam\_checker is called within main
3. spam\_score is called within main

**Link to your repository:** <https://github.com/jace-walker/COP2073>