**LAB #2 – Menu-Driven Python Application Documentation**

Name - Herman Mann

Course – SDEV 300

Date - 03/29/2022

Table of Contents

[**Test Plan** 3](#_Toc98864844)

[**Test Case # 1 - Checking of the generating secure password menu option** 3](#_Toc98864845)

[**Test Case # 2- Checking to see if number gets calculated and formats a percentage** 5](#_Toc98864846)

[**Test Case # 3 – Checking to see the total days until we reach the date July 4, 2025** 6](#_Toc98864847)

[**Test Case # 4 – Checking to see of using the law of cosines to calculate the leg of a triangle** 7](#_Toc98864848)

[**Test Case # 5- Checking to see the accurate calculation of the volume of a right circular cylinder** 8](#_Toc98864849)

[**Test Case # 6 – Checking to see if program exits successfully** 10](#_Toc98864850)

**Pylint results for command-line menu driven python application…………………………….………………………………………………………………**11

# **Test Plan**

|  |  |
| --- | --- |
| **Test Case # 1 - Checking of the generating secure password menu option** | |
| **Input** | a  6  y  y  n  y |
| **Expected Output** | Your Secure Password is: !=Gc@w |
| **Actual Output** | Your Secure Password is: !=Gc@w |
| **Pass?** | **YES** |
| **Screenshots** | Text  Description automatically generatedText  Description automatically generated |

|  |  |
| --- | --- |
| **Test Case # 2- Checking to see if number gets calculated and formats a percentage** | |
| **Input** | b  -1  5  15  4 |
| **Expected Output** | Your Percentage is: 33.3333% |
| **Actual Output** | Your Percentage is: 33.3333% |
| **Pass** | **Pass** |
| **Screenshots** | Text  Description automatically generated |

|  |  |
| --- | --- |
| **Test Case # 3 – Checking to see the total days until we reach the date July 4, 2025** | |
| **Input** | c |
| **Expected Output** | The days until July 4, 2025 is: 1192 |
| **Actual Output** | The days until July 4, 2025 is: 1192 |
| **Pass** | **YES** |
| **Screenshots** |  |

|  |  |
| --- | --- |
| **Test Case # 4 –** | |
| **Input** | d  5  12  13 |
| **Expected Output** | The length of leg C is: 7.752830852403298547648834399 |
| **Actual Output** | The length of leg C is: 7.752830852403298547648834399 |
| **Pass** | **YES** |
| **Screenshots** | Text  Description automatically generated |

|  |  |
| --- | --- |
| **Test Case # 5- Checking to see the accurate calculation of the volume of a right circular cylinder** | |
| **Input** | e  7  10 |
| **Expected Output** | The Cylinder's Volume is: 1539.380400258998626839002100 |
| **Actual Output** | The Cylinder's Volume is: 1539.380400258998626839002100 |
| **Pass** | **YES** |
| **Screenshots** | **Text  Description automatically generated** |

|  |  |
| --- | --- |
| **Test Case # 6 – Checking to see if program exits successfully** | |
| **Input** | f |
| **Expected Output** | Thank you for trying out this application. The program will exit/terminate |
| **Actual Output** | Thank you for trying out this application. The program will exit/terminate |
| **Pass** | **YES** |
| **Screenshots** | **Text  Description automatically generated** |

Pylint results for Menu-Driven Application

Throughout the completion of Lab 2 which focused on developing a command line menu-driven python application providing users with the ability to perform several math and security related functions, I got to learn and experience more in depth knowledge of the Python programming language. I got to experience working with the Python for loop, the while loop in broader context and as well as utilizing the if, elif, and the many control-flow logical statements needed to develop this menu application overall. Discussing the Pylint results of this overall program, I had a lot of issues with “trailing whitespace”, the “function docstring” for describing about the usage of each of the functions in the creation of this overall menu application, and some warnings about not using some imported packages and other import statements and things like that. Luckily towards the end, after running the code analysis I slowly and slowly solved all the issues which were causing these errors for my Pylint score to be better thereafter. I was at a “7.33/10” when I first tested my code, and then by looking at the conventional, warning, and major error issues I was then able to bring my score to a “10.00/10”. The Pylint results was much better compared to my lab 1 results at first glance when I tested lab 1’s Pylint results. Also, this lab made me realize that I have a more better knowledge scope of tackling the Pylint results more faster for future labs.

Graphical user interface, application

Description automatically generated