Sairam Soundararajan

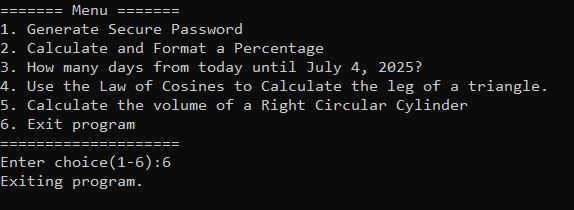
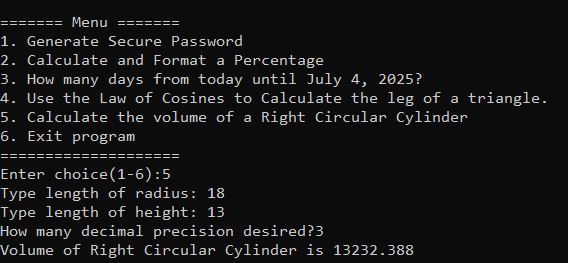
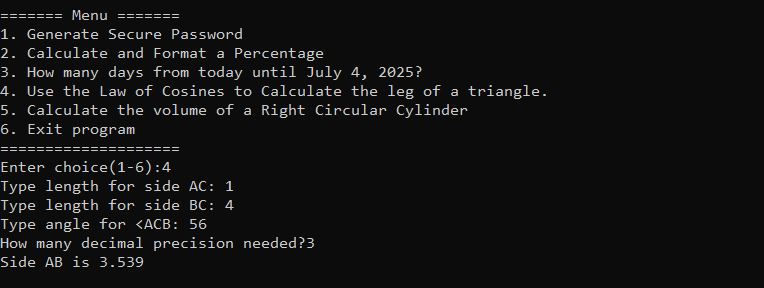
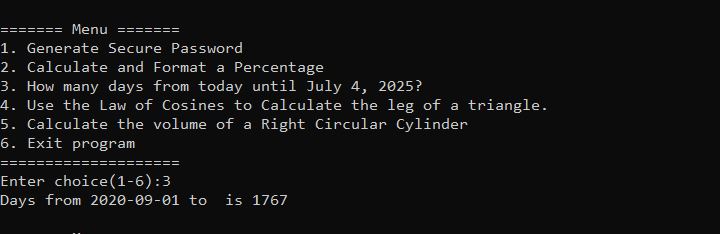
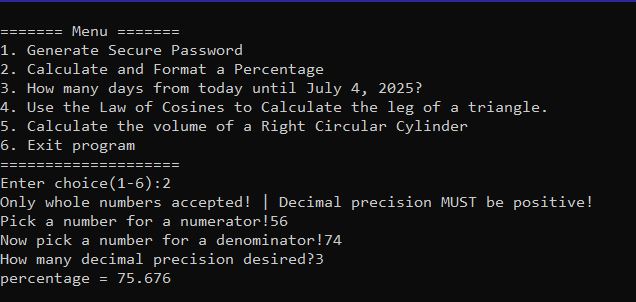
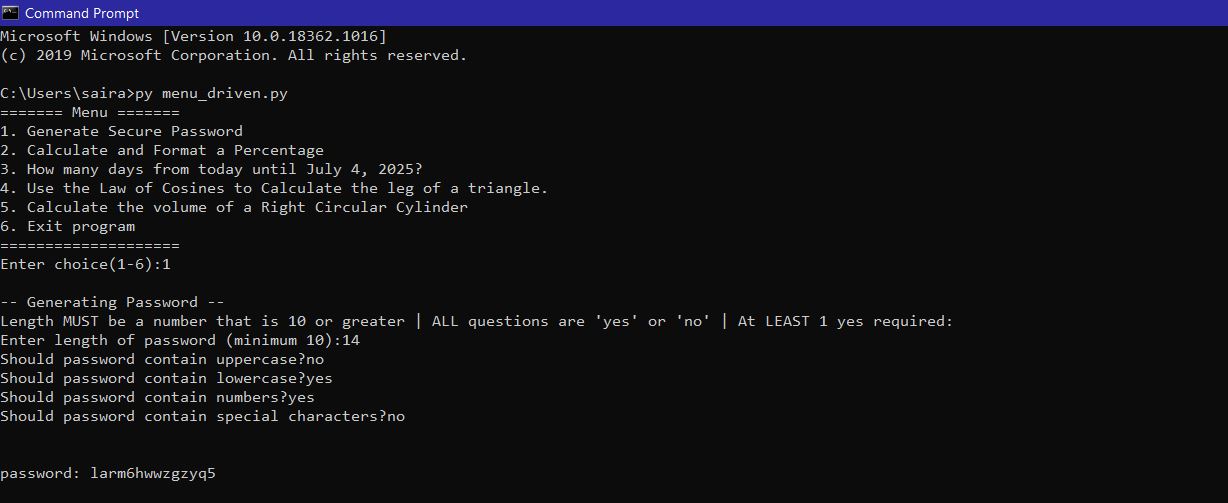
SDEV300: Building Secure Python Applications

University of Maryland Global Campus

Professor Howards

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case | Input | Expected Output | Actual Output | Pass? |
| 1 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):1  -- Generating Password --  Length MUST be a number that is 10 or greater | ALL questions are 'yes' or 'no' | At LEAST 1 yes required:  Enter length of password (minimum 10):14  Should password contain uppercase?no  Should password contain lowercase?yes  Should password contain numbers?yes  Should password contain special characters?no | Password: larm6hwwzgzyq5 | Password: larm6hwwzgzyq5 | Yes |
| 2 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):2  Only whole numbers accepted! | Decimal precision MUST be positive!  Pick a number for a numerator!56  Now pick a number for a denominator!74  How many decimal precision desired?3 | percentage = 75.676 | percentage = 75.676 | yes |
| 3 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):3 | Days from 2020-09-01 to is 1767 | Days from 2020-09-01 to is 1767 | Yes |
| 4 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):4  Type length for side AC: 1  Type length for side BC: 4  Type angle for <ACB: 56  How many decimal precision needed?3 | Side AB is 3.539 | Side AB is 3.539 | Yes |
| 5 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):5  Type length of radius: 18  Type length of height: 13  How many decimal precision desired?3 | Volume of Right Circular Cylinder is 13232.388 | Volume of Right Circular Cylinder is 13232.388 | Yes |
| 6 | ======= Menu =======  1. Generate Secure Password  2. Calculate and Format a Percentage  3. How many days from today until July 4, 2025?  4. Use the Law of Cosines to Calculate the leg of a triangle.  5. Calculate the volume of a Right Circular Cylinder  6. Exit program  ====================  Enter choice(1-6):6 | Exiting program. | Exiting program. | Yes |

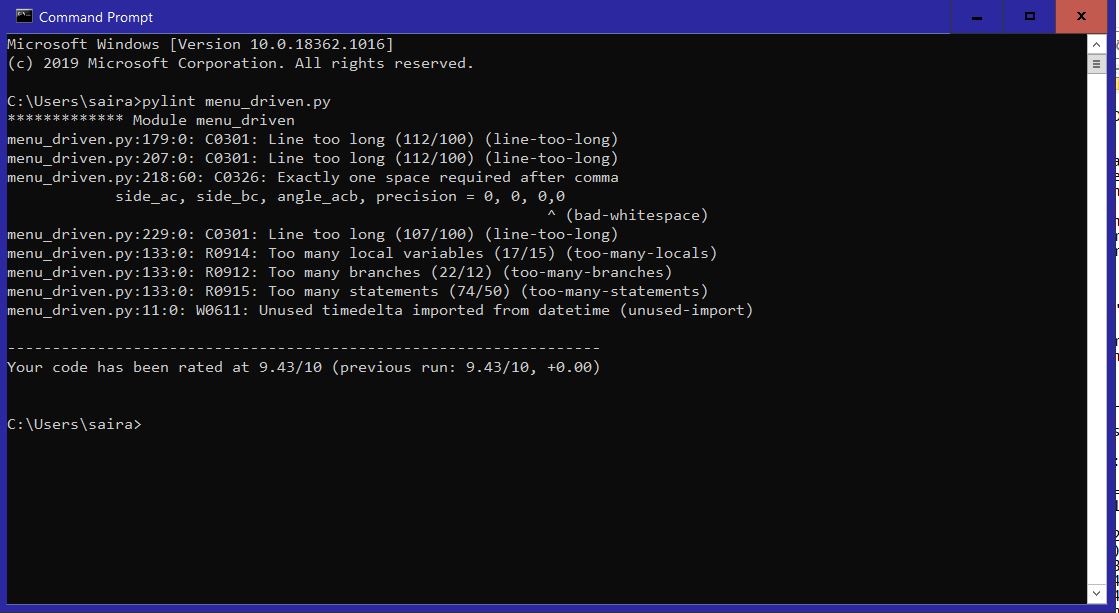
Snapshots of Test Run:



Pylint Discussion and Snapshots:

After successfully finishing up the python code, I ran Pylint for the first time and it rated my code at 9.43/10 because of six conventions and three refactors. The conventions were due to very long lines in three places, needing to put one space between the comma and zero on the line of variable declaration, a bad whitespace, and an unused timedelta from datetime. These conventions were undoubtedly not hard to fix and took little time. The refactors were too many local variables, too many branches, and too many statements. All of these refactors exist in the main method. I did not comply with the refactors because it would have been additional and superfluous work of moving parts of the code around into different functions, which would have made the process unnecessarily longer. The refactors were rather convoluted and would have been ineffective to work on if I have taken that chance. Therefore, when I ran Pylint the second time after fixing those conventions, my code was rated at 9.79/10.

**Before:**



**After:**

