Code Explanation:

Task # 01:

Part a:

I declared a variable x of type double and initialized it with a value of 26. Then we entered whos command. whos command shows current variables in the workspace, their size and type. After that we successively converted y into type int(8-bit, 16-bit,32-bit and 64-bit, signed and unsigned).

Part b:

I declared a variable of type string and then stored my name in it. Then I converted it into an 8-bit integer using built-in int8 function.

Part c:

I converted the output returned by clock into an 8-bit integer. clock returns a six element date vector containing the current time and date in decimal form.

Task # 02:

When this .m file runs , it asks to input two variables, which I store in a and b. Next, I calculate two expressions, L and R, using the input values of a and b. L is simply the sine of the sum of a and b, while R involves the product of sine and cosine values of a and b. Once I have calculated L and R, I used an if statement to check whether they are equal. If they are, I print the message "LHS = RHS, Hence Proved". If not, I print the message "Not Proved". In addition to performing these calculations, I also measure the execution time of the code using the tic and toc functions. This allows the user to see how long it took for me to perform my calculations.

For all trigonometric identities parts(5 in this case), the logic is same. Only the expression of L and R are different.

Task # 03:

Firstly, I prompted the user to enter the credit hours and grade points for each course taken during the semester. This code uses these inputs to calculate the semester GPA by adding up the total credit hours and grade points earned in all courses and dividing the grade points by the credit hours. It also displays calculated cumulative GPA by computing the average grade points for the previous semesters and multiplying that with total credit hours and dividing by the total credits of all previous semesters. Once the GPA is calculated, the code displays a transcript of all courses taken during the semester, including the course names, credit hours, and grade points. The semester GPA is also displayed at the end of the transcript.

Task # 04:

First, I clear the command window and all the workspace variables. Then, I use the clock function to record the start time of the code. Next, I prompt the user to enter two variables and store them in Var1 and Var2. I then perform the swapping of values using a common technique of using arithmetic operations. The value of Var1 is updated to the sum of Var1 and Var2. The value of Var2 is updated to the difference between the new Var1 value and the original Var2 value. Finally, the value of Var1 is updated to the difference between the new Var1 value and the new Var2 value. I then use the disp function to print the swapped values of Var1 and Var2. Lastly, I record the end time using the clock function again and calculate the total time taken by the code using the etime function.

Task # 05:

This code calculates the length of the hypotenuse of a right-angled triangle using the Pythagorean theorem. It prompts the user to enter the length of the perpendicular and the base of the right-angled triangle using the input function and stores them in the variables a and b, respectively. Then, the code calculates the square of the hypotenuse using the formula c^2 = a^2 + b^2, where c is the hypotenuse. It does this by adding the square of the perpendicular to the square of the base using the + and * operators and storing the result in the variable c. Finally, the code displays the length of the hypotenuse using the disp function and the variable c.

Task # 06:

This code prompt the user to enter temperature in Fahrenheit scale and then store it in a variable named Temp_F. Then it convert this value into Celsius scale by using the formula:

$$Tc = \frac{Tf - 32}{1.8}$$

After performing calculation based on above formula, the resulting value is stored in another variable Temp_C and then it is displayed using disp function.

Task # 07:

Next, the user is prompted to input 10 numbers using the input function and storing them in the variable x. The min and max functions are then used to find the smallest and largest numbers in x, which are stored in the variables mini and maxi, respectively. Finally, the code applies the min-max normalization formula to each element in x and displays the resulting values using the disp function. The min-max normalization formula (x-mini)/(maxi-mini) rescales each element in x so that it falls between 0 and 1, with the smallest element in x becoming 0 and the largest element in x becoming 1.

Conclusion:

In conclusion, this lab is designed to provide students with an introduction to MATLAB, a powerful programming language and environment used for numerical computing, data analysis, and visualization. By the end of the lab, I was be able to navigate the MATLAB environment, use built-in mathematical functions, perform variable arithmetic, prompt for input and display output, measure execution time, and create and use M-files.