Calculator:

The task is to create a simple command-line **Calculator** program that performs basic arithmetic operations like **Addition**, **Subtraction**, **Multiplication**, and **Division**. The calculator should handle user inputs, execute the respective operations, and display the results. Additionally, the program should be able to handle various edge cases such as invalid inputs, non-numeric values, and division by zero. It should allow the user to perform multiple calculations in a session until they choose to exit.

Functional Requirements:

1. **Arithmetic Operations:**
    - **Addition**: The user will provide two numbers, and the program will return the sum.
    - **Subtraction**: The user will provide two numbers, and the program will return the difference.
    - **Multiplication**: The user will provide two numbers, and the program will return the product.
    - **Division**: The user will provide two numbers, and the program will return the quotient. If the divisor is zero, it will return an error message.
2. **User Input Validation:**
    - The user will choose one of the four arithmetic operations (A, S, M, or D).
    - The program should handle invalid menu selections and prompt the user to select a valid option if the input is incorrect.
    - The user should input numeric values for the operands. If a non-numeric value is entered, the program will prompt the user to input numbers again.
3. **Error Handling:**
    - If the user tries to divide by zero, the program will return an error message: "Cannot divide by zero!".
    - The program should provide appropriate error messages if the user makes invalid choices or enters non-numeric inputs.
4. **Repeated Calculations:**
    - After completing a calculation, the user will be prompted to either continue with another calculation or exit the program.
5. **Exit Mechanism:**
    - The user can exit the program by typing no, n, or nope when asked if they want to perform another calculation.

Input:

- The user will interact with the program via a series of prompts for:
    - The operation they wish to perform (Addition, Subtraction, Multiplication, or Division).
    - Two numeric values on which the operation will be performed.

Output:

- The program will output the result of the operation, formatted in a readable manner (e.g., "5.0 + 3.0 = 8.0").
- If an error occurs (e.g., invalid input or division by zero), an appropriate error message will be displayed.

Sample Interaction:

Enter 'A' for Addition

Enter 'S' for Subtraction

Enter 'M' for Multiplication

Enter 'D' for Division

Enter Choice (A, S, M, D): A

Enter first number: 10

Enter second number: 5

10.0 + 5.0 = 15.0

Want to do another calculation? (yes/no): yes

Enter 'A' for Addition

Enter 'S' for Subtraction

Enter 'M' for Multiplication

Enter 'D' for Division

Enter Choice (A, S, M, D): D

Enter first number: 10

Enter second number: 0

10.0 / 0.0 = Cannot divide by zero!

Want to do another calculation? (yes/no): no

Goodbye!

Error Scenarios:

1. **Invalid Operation Choice:**
    - If the user enters an invalid operation choice like X or any other character not among A, S, M, or D, the program will prompt them again for a valid choice.
2. **Non-numeric Input:**

- If the user enters a non-numeric value when prompted for numbers, the program will display an error message and ask for valid numeric input.

3. **Division by Zero:**
    - If the user attempts to divide by zero, the program will handle it gracefully by displaying "Cannot divide by zero!".

Edge Cases:

1. Division by zero.
2. Invalid arithmetic operation choices.
3. Non-numeric input for numbers.
4. Multiple calculations in a single session.