Python

import sympy  
  
def parse\_equation(equation\_str):  
 """Parses a mathematical equation string and returns a sympy expression."""  
 return sympy.sympify(equation\_str)  
  
def get\_user\_input(parameter):  
 """Prompts the user for input for a given parameter."""  
 while True:  
 try:  
 value = float(input(f"Enter the value for {parameter}: "))  
 return value  
 except ValueError:  
 print("Invalid input. Please enter a numerical value.")  
  
def evaluate\_equation(equation, parameter\_values):  
 """Evaluates the equation with the given parameter values."""  
 for param, value in parameter\_values.items():  
 equation = equation.subs(param, value)  
 return equation.evalf()  
  
def main():  
 equation\_str = input("Enter the mathematical equation: ")  
 equation = parse\_equation(equation\_str)  
  
 parameters = equation.free\_symbols  
 parameter\_values = {}  
 for param in parameters:  
 value = get\_user\_input(str(param))  
 parameter\_values[param] = value  
  
 result = evaluate\_equation(equation, parameter\_values)  
 print(f"The result of the equation is: {result}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

**Explanation:**

1. **Import the sympy library:** This library provides powerful tools for symbolic mathematics.
2. **Define the parse\_equation function:** This function takes an equation string as input and uses sympy.sympify to convert it into a sympy expression.
3. **Define the get\_user\_input function:** This function prompts the user for input for a given parameter and handles potential input errors.
4. **Define the evaluate\_equation function:** This function substitutes the parameter values into the equation and evaluates it using evalf().
5. **Define the main function:** This is the main entry point of the program. It prompts the user for the equation, identifies the parameters, gets user input for each parameter, evaluates the equation, and prints the result.

**How to use it:**

1. Run the program.
2. Enter the mathematical equation you want to evaluate, using variables for parameters (e.g., "x^2 + 3y").
3. The program will prompt you to enter values for each parameter.
4. Once you enter all the values, the program will calculate and print the result of the equation.

This code provides a flexible and robust solution for parsing and evaluating mathematical equations with user-provided parameters.