A diagram of a algorithm

Description automatically generated with medium confidence

# Main Use Case

Use Case: Evaluate User Equation

Scope: Arithmetic Expression Evaluator

Level: Summary

Context: The goal of the user is to evaluate an arithmetic expression.

Multiplicity: One user will be interacting with one Arithmetic Expression Evaluator. The user may have multiple equations they would like to evaluate, but they will be entered and evaluated sequentially.

Primary Actor: User

# Basic Flow

## User Enters an Equation

### The user types an equation into the system’s textual interface.

## User Submits Their Equation

### The user presses Enter to submit their equation.

## System Evaluates the Equation

### The system parses the user’s equation.

### The system evaluates the parentheses in the equation.

### The system evaluates the operators in the equation.

## System Displays the Result

The system displays the numeric value of the equation to the user.

## Repeat Basic Flow

Restart Basic Flow at **User Enters an Equation**.

# Alternate Flows

## User Does Not Enter an Equation

At the **User Submits Their Equation** step of the ***Basic Flow***, if the user has not entered an equation,

1. Resume basic flow at **Repeat Basic Flow**.

## User Closes the Program

At any step of the ***Basic Flow***, if the user presses the button to close the program,

1. Close the program.
2. End Basic Flow.

## Invalid Symbol Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains an illegal character,

1. Display to the user an error message in the form: “Error: illegal symbol: {offending symbol}!”
2. Resume the Basic Flow at **Repeat Basic Flow**.

## Unmatched Parentheses Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains unmatched parentheses,

1. Display to the user the error message: “Error: unmatched parentheses!”
2. Resume the Basic Flow at **Repeat Basic Flow**.

## Missing Operand Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains an operator that is missing an operand or an empty set of parentheses,

1. Display to the user an error message in the form: “Error: {offending operator} is missing an operand!”
2. Resume the Basic Flow at **Repeat Basic Flow**.

## Operator as Operand Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains an operator whose operand is a non-unary operator,

1. Display to the user the error message: “Error: cannot use operator as operand!”
2. Resume the Basic Flow at **Repeat Basic Flow**.

## Divide by Zero Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains a division operator whose second operand evaluates to zero,

1. Display to the user the error message: “Error: cannot divide by zero!”
2. Resume the Basic Flow at **Repeat Basic Flow**.

## Negative Operand Error

At the **System Evaluates the Equation** step of the ***Basic Flow***, if the user-provided equation contains a modulo or exponentiation operator whose second operand is negative,

1. Display to the user an error message in the form: “Error: {operator symbol}’s second operand cannot be negative!”
2. Resume the Basic Flow at **Repeat Basic Flow**.