You are tasked with building a **Postfix Expression Evaluator** using the **C++ Standard Library stack**. The program should:

1. Accept a postfix mathematical expression (also known as Reverse Polish Notation) as input.
2. Use a **stack** to evaluate the expression.
3. Handle basic arithmetic operations: addition (+), subtraction (-), multiplication (\*), and division (/).
4. Ensure that the evaluation is done correctly by pushing operands onto the stack and applying operators when they appear in the expression.

**Explanation of the Problem:**

1. **Postfix Expression Evaluation:**
   * The postfix expression, also known as Reverse Polish Notation, evaluates arithmetic operations without needing parentheses for precedence rules.
   * Operands are pushed onto a stack until an operator is encountered, at which point two operands are popped, the operation is applied, and the result is pushed back onto the stack.
   * This approach allows for simple evaluation of expressions without the need for complex parsing.
2. **C++ Standard Library Stack:**
   * The program uses the **C++ Standard Library stack (std::stack)** for managing operands during evaluation.
   * **Template-based stacks** allow storing and managing any data type without worrying about memory management or data structure re-implementation, which would be necessary in C.

