# Design of InfixCalculator

The Infix Calculator is designed to evaluate arithmetic expressions written in infix notation. The program supports basic arithmetic operations: addition, subtraction, multiplication, and division. The expressions can include parentheses to indicate the desired order of operations

## Components

### 1. InfixCalculator Class

This is the main class that handles the evaluation of infix expressions. It includes methods for setting the infix expression, converting it to postfix notation, and evaluating the postfix expression. The class uses the LinkedStack class that is given in the textbook for several functions like checking the balance of brackets and expression evaluation.

#### Data Members

* std::string infixExpression: Stores the infix expression provided by the user.

#### Member Functions

* InfixCalculator(): Default constructor.
* bool setInfixExpression(const std::string &expression): Validates and sets the infix expression.
* int evaluate() const: Evaluates the stored infix expression.

#### Private Member Functions

* bool isWellFormed(const std::string &expression) const: Checks if the expression is well-formed.
* bool checkBalancedBrackets(const std::string &expression) const: Checks if the parentheses in the expression are balanced.
* int precedence(char op) const: Determines the precedence of an operator. The operators \* and / have a precedence of 2 and operators + and – have a precedence of 1.
* std::string infixToPostfix(const std::string &infix) const: Converts an infix expression to postfix notation.
* int evaluatePostfix(const std::string &postfix) const: Evaluates a postfix expression.

The class uses the algorithms defined in the textbook to perform fucntions like checking balance of brackets and expression conversion and evaluation.