Concept Explanation: Dynamic Calculator

This document provides a professional overview of the concept, working, and logic behind the given Python program 'dynamic calculator.py'.

The Dynamic Calculator is a command-line Python program that evaluates arithmetic expressions entered by the user. It supports basic operations such as addition (+), subtraction (-), multiplication (\*), and division (/). The program parses the input string manually and performs the operations in the correct mathematical order.

Concept Used

* The program is based on the following concepts:
* String processing and user input handling.
* Use of loops and conditional statements to parse expressions.
* Manual implementation of operator precedence without using Python’s eval() function.
* Dynamic list manipulation to perform step-by-step calculations.

Working Mechanism

1. The process flow of the calculator can be explained in steps:
2. The program continuously prompts the user to enter an expression until 'q' is entered to quit.
3. It parses the input string, separating numbers and operators into a list.
4. Multiplication (\*) and division (/) are processed first to maintain operator precedence.
5. Addition (+) and subtraction (-) are performed afterward.
6. The final result is displayed to the user.

Why This Concept is Used

* The manual parsing approach is used instead of built-in evaluation functions like eval() for the following reasons:
* To demonstrate how arithmetic operations can be implemented logically using programming fundamentals.
* To ensure better control over operation order and custom functionality.
* To avoid security risks that come with using eval(), which can execute arbitrary code.

Key Features

* Handles basic arithmetic operations with correct precedence.
* Uses loops and indexing to simulate arithmetic computation.
* Simple and interactive command-line interface.