

## **ABSTRACT**

The Basic Arithmetic Expression Compiler project aims to develop a simple yet efficient compiler for basic arithmetic expressions. The compiler takes mathematical expressions as input and translates them into executable machine code or an intermediate representation for further processing. The primary goal is to explore fundamental concepts of lexical analysis, syntax parsing, and code generation within the context of arithmetic expressions.

The project involves designing and implementing a lexer to break down input expressions into tokens, a parser to analyze the syntactic structure of the expressions, and a code generator to produce executable code. Emphasis is placed on supporting standard arithmetic operations such as addition, subtraction, multiplication, and division. Additionally, the compiler may include features like variable assignment and basic control flow constructs to enhance its functionality.

By creating a Basic Arithmetic Expression Compiler, the project aims to provide a hands-on learning experience for individuals interested in compilers and programming language implementation. The simplicity of the compiler allows for a clear understanding of the core principles involved in translating high-level mathematical expressions into machine-executable instructions.