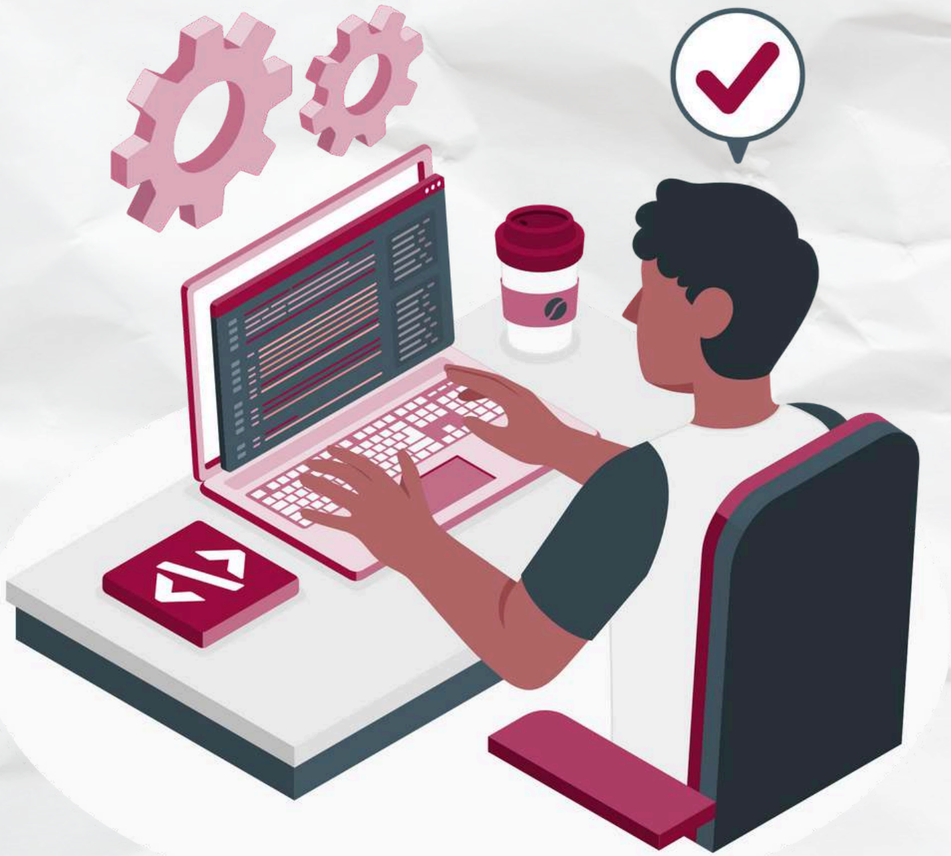
COMPILER DESIGN PROJECT



DR.Heba Elhadidi

OUR TEAM

Samaa

Nayra

Sabrin

Ahmed

INTRODUCTION

- **Project goal**:

To build a compiler for a simple programming language.

- **Project components**:

Lexical analysis, symbol table creation, FIRST and FOLLOW sets calculation, parsing tree generation, parsing table generation.

THE COMPILER STRUCTURE

BNF Grammar

Lexical analysis

Symbol Table

FIRST/FOLLOW Sets

Parse Tree

handle error

GRAMMAR

- **Definition:**

Rules that define the syntax of the programming language, like variable declarations, expressions, conditional statements.

- **Grammar Example:**

Rules for variables (var_decl), mathematical expressions (expr), conditional statements (if_stmt), etc.

LEXICAL ANALYSIS

- **Definition:**

Extracting tokens from the code using regular expressions (regex).

- **Token Patterns:**

Keywords, variables, numbers, strings, arithmetic and logical operators, etc.

- **Method:**

Using regular expressions to analyze the code and identify the tokens.

SYMBOL TABLE

- **Definition:**

A table that stores information about variables such as type, address, declaration line number, and usage.

- **Method:**

Analyzing the code to extract variables and storing their details such as type and address.

FIRST AND FOLLOW SETS CALCULATION

- **FIRST Sets:**

Determine which symbols can appear first in the expansions.

- **FOLLOW Sets:**

Determine which symbols can follow others in the expansions.

- **Method:**

Using rules to compute these sets for each non-terminal.

PARSE TREE

- **Definition:**

A data structure that represents the syntactic analysis of the program.

- **Method:**

Constructing a tree that represents the source code based on the defined grammar, such as code for variable declarations or conditional statements.

ERROR HANDLING

- **Purpose:**

Detect and handle errors in source code files efficiently.

- **Features:**

Missing Semicolons, Brace Matching Errors, Data Type Errors and Variable Naming Errors

USER INTERFACE

- **Goal:**

Provide a graphical interface to display analysis results.

- **Features:**

Display tokens, symbol table, FIRST and FOLLOW sets, parse tree, and parse table.

- **Method:**

Using Tkinter and graphical interfaces to display results in an organized manner.

CONCLUSION

- **Achievements:**

A compiler was built that supports lexical analysis, symbol tables, FIRST and FOLLOW sets, and parse tree generation.

- **Challenges:**

The challenges faced during development, such as error handling and parsing complex rules.

- **Next Steps:**

Improving the code and adding further optimizations.

THANK YOU