T-603-THYD Compilers

Project: A *parser* for ChocoPy

Objective

The objective of this project is to get you started with writing a compiler/interpreter. The focus of this part of the project is on syntax analysis.

You should work on the project in a group of two. You need the instructor's permission to do the project individually.

Description

Your task is to write a recursive-descent parser for the ChocoPy programming language (in Python). Please study section four in the *ChocoPy language reference* carefully (found in Canvas/Modules/Project) to get a complete understanding of the tokens your lexer needs to recognize. We will follow the specifications there faithfully (except that you should return token STRING in place of IDSTRING).

You have been provided with an incomplete skeleton code for the parser in Python. You are free to modify it as you see fit; however, you are not allowed to change the interface to the parser.

I recommend that you write the parser using the following steps:

1. Rewrite the grammar for ChocoPy to be suitable for a recursive-descent parser. Feel free to use EBNF notation if you prefer.
2. Write your parser without worrying about generating the *abstract syntax tree (AST).*
3. Add the AST generation part to your parser.

In *Canvas*, submit your *Python parser code, the test cases you used for testing your parser, and a file with your grammar description*, more specifically:

- A Python file called *parser.py.*
- One or more text files with your test cases (using any names you see fit).
- A PDF of a text file with your grammar description.