

Name: Kaumil Patel

UCID: 30088096

Email: [kaumil.patel@ucalgary.ca](mailto:kaumil.patel@ucalgary.ca)

CPSC Gitlab: <https://gitlab.cpsc.ucalgary.ca/kaumil.patel/cpsc-411>

Self Assessment: 6/8

Feedback:

I have tried to adhere to the recommended practises listed in the bison documentation. But I was curious if I should further subdivide my ast into sub classes. Please advise me if I need to add more nodes to my AST.h file.

Evidence:

I have revised my self assessment score to reflect the fact that the for loop condition is not displayed when it is not provided. This is a minor error in the print function in the for node and is an easy fix. Therefore, the underlying parser I believe has good performance.

1. Tool (milestone) properties.

- Example error and warning messages output.
  - 1. Error: Unexpected token {=} at line 7
- Example output - parse.t5
  - 1. Program
  - 2.       func at line 3
  - 3.           new id [main] at line 3
  - 4.           sig
  - 5.           formals
  - 6.           id type [void]
  - 7.       Block
  - 8.           if at line 5
  - 9.               == at line 5
  - 10.               + at line 5
  - 11.                   number [2] at line 5
  - 12.                   number [3] at line 6
  - 13.                   number [5] at line 7
  - 14.           Block
  - 15.           empty stmt

2. Code qualities.

- Each code block has a comment explaining what it does. To make it simpler to read, the code is also divided into sections.
- Depending on its intended use, the output is correctly sent to either stderr or stdout.
- I have verified that the code passes the provided tests as well as my own tests to show that it is compliant and error-free. The results mostly line up with the reference parser.
- To make the implementation simpler, I am using flex and bison.
- In order to make the code modular and easily extendable, it is divided into functions and objects.
- Compilation output:
  1. `kaumil.patel@csx2:~/cpsc 411$ make`
  2. `flex --c++ -o scanner.cc scanner.l`
  3. `bison parser.yy -o parser.cc -Wall`
  4. `g++ scanner.cc parser.cc main.cpp -o golf -Wall -Wextra`
  5. `kaumil.patel@csx2:~/cpsc 411$`