

King Saud University
College of Computer and Information Sciences
Computer Science Department
CSC 340: Programming Language and Compilation
Programming Assignment 2 & 3: Using JFLex & BYACC/J

For this assignment, you will use JFLex for the lexical part of the task and BYacc/J for writing a parser for a simple language described as:

$P \rightarrow D ; P \mid D$
 $D \rightarrow \text{def id}(\text{ARGS}) = E ;$
 $\text{ARGS} \rightarrow \text{id} , \text{ARGS} \mid \text{id}$
 $E \rightarrow \text{int} \mid \text{id} \mid \text{if } E1 \text{ OP } E2 \text{ then } E3 \text{ else } E4 \mid \text{repeat } E1 \text{ until } E2$
 $\quad \mid E1 + E2 \mid E1 - E2 \mid E1 * E2 \mid E1 / E2 \mid \text{id}(E1, \dots, En)$
 $\text{OP} \rightarrow == \mid > \mid < \mid >= \mid <= \mid !=$

Here is a simple program for computing Fibonacci numbers, written in the language

```
def fib(x) = if x = 1 then 0 else
            if x = 2 then 1 else
            fib(x - 1) + fib(x - 2)
```

Part I: Lexical Analysis of Mython (Assignment-2)

You will use JFLex to capture the tokens of the language. To do this, you need to define a constant value associated with each token, so that you can return this value every time your parser needs a new token.

- **Keywords:** def, if, then, else, repeat, until.
- **Punctuation elements:** () : , ;
- **Operators:** = , + , - , * , / , == , > , < , <= , >= , !=
- **Identifiers:** String of alphanumeric (and _) starting with an alphabetic character
- **Literals:** integer only

Hint:

There are lots of online resources on JFLex including tutorials on youtube. See for example <https://www.youtube.com/watch?v=IV1Rwq7ERR4>

- 1) This assignment can be group of 2 students
- 2) Submit your pdf file and java file in **one zip file on LMS (Make sure only ONE student of each group submits)**
- 3) Your pdf file should contain:
 - i. A cover page (your name, serial number and section number)
 - ii. Java code and screen shots of your code and outputs
- 4) Due date (Monday-18 March 2024 /23:59 PM)

Part II: The parser (Assignment-3)

- 1) Rewrite the grammar to eliminate any ambiguity and make sure that the associativity of the operations + , − , * , and / are left to right.
- 2) Write a parser of the rewritten grammar using BYACC/J. Once you have built your parser, you should be able to process programs written in the language described above. When BYACC finds input that doesn't match the grammar, it automatically terminates with the message 'Syntax error'. You will need to write an error routine (yyerror) that also prints out the line number before this termination, and a main method that reads a program from a file.

The above program should compile and have no error

Hint: Also, there are lots of online resources on BYACC/J (see for example <http://byaccj.sourceforge.net/>) and video tutorials.

- 1) This assignment can be a group of 2 students

Submit your pdf file and java file in **one zip file on LMS (Make sure only ONE student of each group submits)**

- 2) Your pdf file should contain:

- i. A cover page (your name, serial number and section number)
- ii. Java code and screen shots of your code and outputs

- 3) Due date ()