**CPCS-302 (Sections CS1 & CS3)**

**Group Project Part 2 (Recursive Descent predictive parser):**

|  |  |
| --- | --- |
| **CLO: 8** | **SO: 2** |

**Assigned Date: Monday October 18, 2024**

**Due Date: Thursday November 07, 2024 till 11:59 Mid Night**

Consider the following CFG which generates the language of all infix arithmetic expressions:

E T E’

E’ + T E’ | – T E’ | ɛ

T F T’

T’ \* F T’ | / F T’ | ɛ

F ( E) | id

Write a complete Java program to construct recursive descent predictive parser for the above grammar, to check the syntax of all arithmetic expressions involving digits separated with the operators +, –, \*, / and ( ), stored in a file.

**Implementation Restrictions:**

1. You should read the arithmetic expressions from the **file “input.txt”** and generate the **outputs** for all these expressions on **screen**.
2. As soon as parser finds some syntax error, it should display the message “Syntax Error” and exit that input (and parse the remaining inputs).
3. Only **one member** of the group should upload the solution on Blackboard.
4. Plagiarism and cheating will result in 0 marks.
5. Upload complete project on Blackboard.
6. Solutions uploaded in any other format (like .docx, .pdf etc will not be accepted)

**Sample input and output**

**If the input file (input.txt) contains the following arithmetic expressions (tokens are separated with spaces):**

id \* ( id + id ) $

( id + id ) \* id $

( id \* + id $

id + id \* id $

**the output displayed on screen should be as follows:**

Correct Syntax

Correct Syntax

Syntax Error

Correct Syntax

**Important Note:**

**Group Project Part 2** has been uploaded in “Group Project” option of Blackboard. Its due date is **Thursday November 07, 2024 till 11:59 PM**. You can also upload the solution by **Friday November 08, 2024 till 11:59 PM with 25% deduction of marks and by Saturday November 09, 2024 till 11:59 PM with 50% deduction of marks**. **After this date, no submission will be accepted.**