Assignment 03

Compiler

Group _04

Team Members: Tejendra Khatri, Sushil Pandey, Ankur Lamichhane

Purpose of the assignment: To build the abstract syntax tree of the input program.

All the requirements given in the assignment is completely done. Class Lexer reads the input from the text file and class parser extends ASTVisitor. Parser then generate the abstract syntax tree based on the input that Lexer read from textfile. Necessary node classes are added in parser file directory. PreetyPrinter is also implemented and if input program doesn't follow the programming convention like indentation. Then your pretty printer will print your program on out file (in output.txt).

Roles and Responsibilities:

As this program was little tough, we studied separately at first about the visitor's pattern and discussed the ideas to start with. Many times, we stayed together and share the progress that we all achieve. Tough we all involve in coding, Tejendra played vital in coding role and Sushil, and Ankur, helped inside coding and debugging along with grammar and pdf file. So, all have played significant role in the completion of the project.

Grammar that we design:

Group_04_Compiler_Assign03_Grammar:

CompilationUnit Node ➔BlockNode

BlockNode➔ {DeclarationNode StatementNode}

DeclarationNode ➔ DeclarationNode | e

DeclarationNode➔TypeNode LiteralNode | e

StatementNode➔AssignmentNode | e

AssignmentNode➔Literal = OperatorNode

OperatorNode➔ AdditionNode | SubtractionNode |
MultiplicationNode | DivisionNode | ValueNode

AdditionNode➔LiteralNode + LiteralNode

SubtractionNode➔LiteralNode – LiteralNode

MultiplicationNode➔LiteralNode *LiteralNode

DivisionNode➔LiteralNode /LiteralNode