**Grammar for Monkey Language:**

1. program : statement\*
2. statement : **let** letStatement | **return** returnStatement | expressionStatement
3. letStatement : **identifier =** expression ( **;** | ε)
4. returnStatement : expression ( **;** | ε)
5. expressionStatement : expression ( **;** | ε)
6. expression : additionExpression comparison
7. comparison : ((**<**|**>**|**<=**|**>=**|**==**) additionExpression)\*
8. additionExpression : multiplicationExpression additionFactor
9. additionFactor : ((**+**|**-**) multiplicationExpression)\*
10. multiplicationExpression : elementExpression multiplicationFactor
11. multiplicationFactor : ((**\***|**/**) elementExpression)\*
12. elementExpression : primitiveExpression (elementAccess | callExpression)
13. elementAccess : **[** expression **]** | ε
14. callExpression : **(** expressionList **)** | ε
15. primitiveExpression : **Integer** | **String** | **identifier** | **true** | **false** | **(** expression **)** | arrayLiteral | arrayFunctions **(** expressionList **)** | functionLiteral | hashLiteral | printExpression | ifExpression
16. arrayFunctions : **len** | **first** | **last** | **rest** | **push**
17. arrayLiteral : **[** expressionList **]**
18. functionLiteral : **fn** **(** functionParameters **)** blockStatement
19. functionParameters : **identifier** moreIdentifiers
20. moreIdentifiers : (**, identifier**)\*
21. hashLiteral : **{** hashContent moreHashContent **}**
22. hashContent : expression **:** expression
23. moreHashContent : (**,** hashContent)\*
24. expressionList : expression moreExpressions | ε
25. moreExpressions : (**,** expression)\*
26. printExpression : **puts** **(** expression **)**
27. ifExpression : **if** expression blockStatement(**else** blockStatement | ε)
28. blockStatement : **{** statement\* **}**