**Tiny Language Context Free Grammar**

1. Program:

Program -> Functions Main\_Function

1. Main\_Function:

Main\_Function -> DataType main () Function\_Body

1. Functions:

Functions -> Functions Function | **ε**

* Functions -> Function Functions’ | **ε**
* Functions’ -> Function Functions’ | **ε**

1. Function:

Function -> Function\_Declaration Function\_Body

1. Function\_Body:

Function\_Body -> {Statements Return\_Statement}

1. Statements:

Statements -> Statements Statement | Statement | **ε**

* Statements -> Statement Statements’
* Statements’ -> Statement Statements’ | **ε**
* **Statement -> Assignment\_Statement; |**

**If\_Statement |**

**Declaration\_Statement |**

**Write\_Statement |**

**Read\_Statement |**

**Return\_Statement |**

**Repeat\_Statement**

**Function\_Call | ε**

1. Function\_Declaration:

Function\_Declaration -> Datatype Function\_Name (**ε | Parameters**)

1. Parameters:

Parameters -> Parameter (Multi\_Parameter | **ε)**

* Multi\_Parameter -> ,Parameter Multi\_Parameter

1. Parameter:

Paremeter -> DataType Identifier

1. FunctionName:

Identifier

1. Repeat\_Statement:

Repeat\_Statement -> repeat Statements until Coniditon\_Statement

1. If\_Statement:

If\_Statement -> if Condition\_Statement then Statements (Else\_If\_Statements | Else\_Statement **| end)**

1. Else\_If\_Statements:

Else\_If\_Statements -> elseif Condition\_Statement then Statements (Else\_If\_Statements | Else\_Statement **| end)**

1. Else\_Statement:

Else\_Statement -> else Statements end

1. Condition\_Statement

Condition\_Statement -> Condition (**ε** | (Boolean\_Operator Condition\_Statement))

1. Condition

Condition -> Identifier Condition\_Operator Term

1. Return\_Statement

Return\_Statement -> return Expression;

1. Read\_Statement:

Read\_Statement - > read Identifier;

1. Write\_Statement:

Write\_Statement - > write Identifier;

1. Declaration\_Statement:

Declaration\_Statement -> DataType (Assignment\_Statement | Identifer) Multi\_Variable

* **Multi\_Variable -> ,(** Assignment\_Statement | Identifer**) Multi\_Variable | ε**

1. Data\_Type:

Data\_Type -> int | float | string

1. Assignment\_Statement:

Assignment\_Statement -> Identifier := Expression

1. Expression:

Expression -> string | Term | Equation

1. Term:

Term -> Number | Identifier | Function\_Call

1. Function\_Call:

Function\_Call -> Identifier (**ε | (Identifer Identifiers)**)

* Identifiers -> (,Identifer Identifiers) | **ε**

1. Equation:

Equation -> Term\_before\_operator

(Term | (\(Equation\) Terms\_after\_operator))

* Terms\_after\_operator -> **ε** | (Arithmetic\_Operator Term Terms\_after\_operator)
* Terms\_before\_operator -> **ε | (Term Arithmetic\_Operator Terms\_before\_operator)**

1. Arithmetic\_Operator

Arithmetic\_Operator -> + | - | \* | /

1. Condition\_Operator

Condition \_Operator -> = | > | < (> | **ε)**