

|  |  |
| --- | --- |
| Faculty of Computer & Information Sciences  Ain Shams University  Subject: CSC360  Compilers’ Theory | Examiners: Prof. Zaki Taha  Dr. Sally Saad  Academic year: 1st term 2021-2022  Year: 3ed undergraduate |

**Compilers’ Theory**

**Milestone -1-**

1. **The “TINY” Language Regular Expressions:**
2. Number:

Digit := [0-9]

Num\_Un\_Signed := (Digit)+

Num\_Signed := (+|-)? Num\_Un\_Signed

Num\_float := Num\_Signed (\.Num\_Un\_Signed)?

1. String:

Letter = [a-z]|[A|Z]

Str := ^\".\*\"$

1. Reserved\_Keywords:

R\_Keywords := int|float|string|read|write|repeat|until|if|elseif|else|then|return|end

1. Comment\_Statement:

L\_Comment := ^/\\*.\*\\*/$

1. Identifiers:

identifier := Letter(Letter|Digit)\*

1. Function\_Call:

Fun\_call := identifier \( ((identifier)(, identifier)\*)? \)

1. Term:

Term := ( Num\_float | identifier | Fun\_call)

1. Arithmatic\_Operator:

Arth\_op = (+ | - | \* | /)

1. Equation:

E\_unit = (Term+ Arth\_op)\*(Term+)$

Equ = E\_unit | (Term Arth\_op)\* \( E\_unit \)( Arth\_op Term)\*

1. Expression:

Exp := Term|Str|Equ

1. Assignment\_Statement:

Ass\_st:= (identifier := Exp)

1. Datatype:

Datatype := (int|float|string)

1. Declaration\_Statement:

Dec\_st := ^Datatype identifier (,identifier|,Ass\_st)\*;$

1. Write\_Statement:

Write\_st:= ^ write (EXp|\n) ;$

1. Read\_Statement:

Read\_st:= ^ read identifier ;$

1. Return\_Statement:

Return\_st:= ^ return Exp ;$

1. Condition\_Operator:

Con\_op:= (<|>|=|<>)

1. Condition:

Con:= (identifier Con\_op term)

1. Boolean\_Operator:

Boolean\_Op:= (&& | ||)

1. Condition\_Statement:

Condition (Boolean\_Operator Condition)\*

1. Set\_of\_Statements

Set\_of\_Statements := (Assignment\_Statement | Declaration\_Statement | Write\_Statement | Read\_Statement | (Return\_Statement)? | Function\_Call)

1. If\_Statement:

If\_Statement := “if” Condition\_Statement “then” Set\_of\_Statements (Else\_If\_Statement | Else\_Statement | end )

1. Else\_If\_Statement:

Else\_If\_Statement := “elseif” Condition\_Statement then Set\_of\_Statements (Else\_If\_Statement | Else\_Statement | “end”)

1. Else\_Statement:

Else\_Statement := ”elseif” Condition\_Statement “then” Set\_of\_Statements (Else\_If\_Statement | Else\_Statement | “end”)

1. Repeat\_Statement:

Repeat\_Statement := “repeat” Set\_of\_Statements “until” Condition\_Statement

1. FunctionName:

FunctionName := Identifier

1. Parameter:

Parameter := Datatype Identifier

1. Function\_Declaration:

Function\_Declaration :=

Datatype FunctionName \( (Parameter(,Parameter)\*)? \)

1. Function\_Body:

Function\_Body := { Set\_of\_Statements (Return\_Statement) }

1. Function\_Statement:

Function\_Statement:= Function\_Declaration Function\_Body

1. Main\_Function:

Main\_Function := Datatype “main“ \( \ ) Function\_Body

1. Program:

Program := (Function\_Statment)\* Main\_Function