# -:Type checker Implementation:-

-Lakshmi Hegde(MT2021072)

## Program Structure:

Program

Function Definition

Block

Variable Declarations

Sequence

Assign statements
Loops
If conditions
Return statements
Function calls

Block (Same str as above)
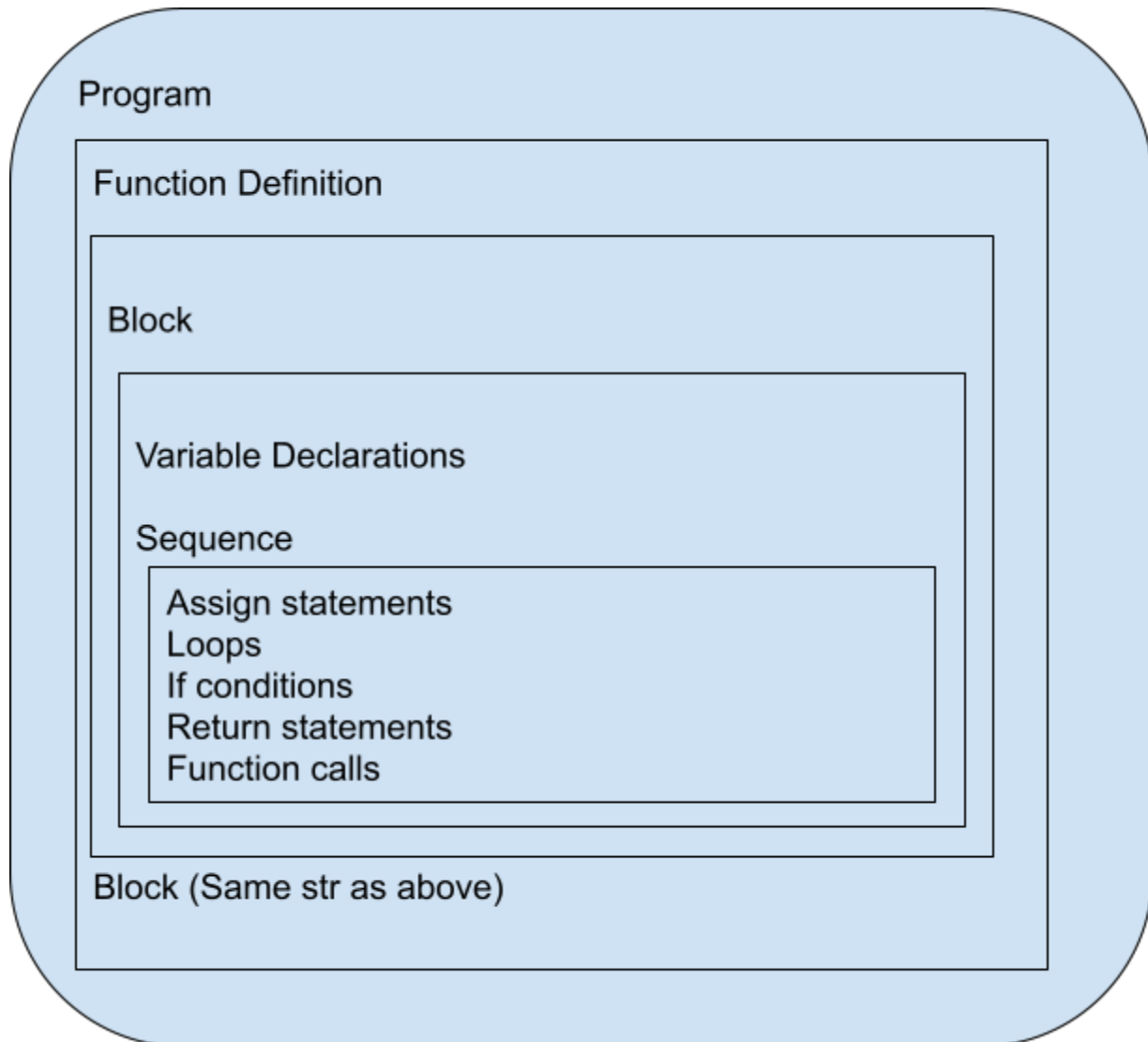
# Requirements :

1. Program must have at least one function definition followed by a block.
2. Declarations inside the block are optional.
3. Block must have at least one instruction.
4. No returns inside loops and if condition.
5. Only num and bool types are implemented. This makes the return statement mandatory inside the function body.
6. Function calls can be used as parameters inside function and also as return statements.
7. Return is considered to be the last statement of the block. Any statements following the return statement are ignored.
8. If any statements are present after return, the implementation just ignores them without type checking but no error is produced .

# Examples:

## 1.basicfeatures.txt

```
num fun(num x,num y)
{
        num z
        z= x * y
        return x+y+z
}

num z
num res
z=10
res=fun(z, z*z)

return res
```

This program covers basic features like

# Function definition

```
num fun(num x,num y)
{
        num z
        z= x * y
        return x+y+z
}
```

# Block

```
num z
num res
z=10
res=fun(z, z*z)
```

return res

# Function call
fun(z, z*z)

# Assign
z=10
res=fun(z, z*z)

# Declarations at local scope
num z
num res
num z inside function fun.

# Expressions
x * y
x + y +z
z * z

# Return
return res

# 2. If_ex.txt

```
num check (num x , bool y)
{
        if y then
                x=x+x
        else
                x=2*x

        return x
}

return check(10,true)
```

## If condition:

```
if y then
        x= x+x
else
        x = 2 * x
```

If conditions can occur as part of blocks.
Only the " if then else" structure is implemented .

# 3. Loop_ex.txt

```
num add(num x,num y, bool test)
{
        num res
        loop(test)
        {
                res= x + y
                test = false
        }
        return res
}

num z
num y
z=10
y=7

loop(true)
{
        num res
        res=add(z,y,true)
        y=z
        z=res
}
```

Loops:

```
loop(test)
{
        res= x + y
        test = false
}
```

```
loop(true)
{
        num res
        res=add(z,y,true)
        y=z
        z=res
}
```
Loops occur as part of blocks.


# 4.recursive_ex.txt

```
num add(num x,num y)
{
        return add(x+y , y*x )
}

bool call()
{
        num z
        z=10

        add( add(z, z+z) ,  -z)

        return call()
}

return call()
```

# Recursive calls:

```
num add(num x,num y)
{
       return add(x+y , y*x )
}
```

"Add" function calls itself.

```
bool call()
{
       num z
       z=10

       add( add(z, z+z) ,  -z)

       return call()
}
```

Here "call" can make a call to the "add" function and also can call to itself.

FORWARD REFERENCE results in error. That's why we cannot call the function "call" inside the function "add". This results in error as per my implementation.

# Output Screenshots:

```
Function "Num  fun ( Num x   Num y    )
{
Num z
z = x * y

return x + y + z
}"
evaluates Num
----------------------------------------------------------------

ASSIGN

z = 10
evaluates to null
----------------------------------------------------------------

EXPRESSION
z * z
evaluates to Num
----------------------------------------------------------------

FUNCTION CALL
fun ( z   z * z   )
evaluates to Num
----------------------------------------------------------------

ASSIGN

res = fun ( z   z * z   )
evaluates to null
----------------------------------------------------------------

RETURN

return res
evaluates to Num
----------------------------------------------------------------

BLOCK
Num z   Num res
z = 10

res = fun ( z   z * z   )

return res

evaluates to Num
----------------------------------------------------------------

Done!
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

```
IF CONDITION
IF ( test ) then res else res * 2
evaluates to Num
```

```
LOOP
LOOP ( true )
{
Num res
res = add ( z    y    true    )

y = z

z = res

}
evaluates to null.
```

# Errors:

## 1. Parameter type mismatch.

```
num add(num x,num y)
{
    return x+y
}

return add(1,true)
```

```
Function call "add ( 1    true   ) " didn't typecheck. Parameter type mismatch.
java.lang.Exception: Function call "add ( 1    true   ) " didn't typecheck. Parameter type mismatch.
        at Typechecker.typecheckFunccall(Typechecker.java:282)
        at Typechecker.typecheckSequence(Typechecker.java:117)
        at Typechecker.typecheckBlock(Typechecker.java:96)
        at Typechecker.typecheckProgram(Typechecker.java:46)
        at Driver.main(Driver.java:22)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

## 2. Number of arguments mismatch

```
num add(num x,num y)
{
    return x+y
}

return add(1)
```

```
Function call"add ( 1   ) " didn't typecheck.Number of arguments mismatch.
java.lang.Exception: Function call"add ( 1   ) " didn't typecheck.Number of arguments mismatch.
        at Typechecker.typecheckFunccall(Typechecker.java:291)
        at Typechecker.typecheckSequence(Typechecker.java:117)
        at Typechecker.typecheckBlock(Typechecker.java:96)
        at Typechecker.typecheckProgram(Typechecker.java:46)
        at Driver.main(Driver.java:22)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

## 3. Return statement inside loop

```
num add(num x,num y)
{

        loop(true)
        {
                return x+y
        }
        return x
}

add(1,4)
```

```
LOOP didn't type check. Presence of return statement
java.lang.Exception: LOOP didn't type check. Presence of return statement
        at Typechecker.typecheckLoop(Typechecker.java:227)
        at Typechecker.typecheckSequence(Typechecker.java:109)
        at Typechecker.typecheckBlock(Typechecker.java:96)
        at Typechecker.typecheckBody(Typechecker.java:78)
        at Typechecker.typecheckFunDeclarationList(Typechecker.java:68)
        at Typechecker.typecheckProgram(Typechecker.java:44)
        at Driver.main(Driver.java:22)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

4. When you violate the structure of the program you can get the parsing time errors like:

```
num x
x = 10
```

This program violates the structure of function declaration followed by block. It skips function declaration and just has one block. This causes a parsing error.

```
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$ java -cp /home
/lakshmi/cup-11a.jar:./ Driver examples/loops.txt
Syntax error at character 1 of input
null
java.lang.NullPointerException
        at java_cup.runtime.lr_parser.symbl_name_from_id(lr_parser.java:456)
        at java_cup.runtime.lr_parser.report_expected_token_ids(lr_parser.java:446)
        at java_cup.runtime.lr_parser.syntax_error(lr_parser.java:433)
        at java_cup.runtime.lr_parser.parse(lr_parser.java:725)
        at Driver.main(Driver.java:18)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

5. Forward reference error

```
num add(num x,num y)
{
        call()
        return add(x+y , y*x )
}
bool call()
{
        num z
        z=10
        add( add(z, z+z) ,  -z)

        return call()
}

return call()
```

```
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$ java -cp /home/lakshmi/jflex-1.8.2/lib/jflex-full-1.8
/lakshmi/cup-11a.jar:./ Driver examples/loops.txt
call Not defined.
java.lang.Exception: call Not defined.
        at Typechecker.typecheckFunccall(Typechecker.java:294)
        at Typechecker.typecheckSequence(Typechecker.java:117)
        at Typechecker.typecheckBlock(Typechecker.java:96)
        at Typechecker.typecheckBody(Typechecker.java:78)
        at Typechecker.typecheckFunDeclarationList(Typechecker.java:68)
        at Typechecker.typecheckProgram(Typechecker.java:44)
        at Driver.main(Driver.java:22)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```

## 6. Undefined variable error

```
num add(num x,num y)
{
        return x + y + z
}

return add(10 , 1)
```

```
Variable z not declared.
java.lang.Exception: Variable z not declared.
        at Typechecker.typecheckIdExpr(Typechecker.java:346)
        at Typechecker.typecheckExpression(Typechecker.java:255)
        at Typechecker.typecheckAddExpr(Typechecker.java:300)
        at Typechecker.typecheckExpression(Typechecker.java:243)
        at Typechecker.typecheckReturn(Typechecker.java:126)
        at Typechecker.typecheckSequence(Typechecker.java:107)
        at Typechecker.typecheckBlock(Typechecker.java:96)
        at Typechecker.typecheckBody(Typechecker.java:78)
        at Typechecker.typecheckFunDeclarationList(Typechecker.java:68)
        at Typechecker.typecheckProgram(Typechecker.java:44)
        at Driver.main(Driver.java:22)
lakshmi@lakshmi-VivoBook-ASUSLaptop-X412FJC-X412FJ:~/Desktop/Final/Final2/Assignment_4$
```