**[TBD] Language**

**KEYWORDS:**

1. @i (Integer)
2. @f (float)
3. @c (Character)
4. @b (Boolean)
5. @i64 (64 bit Integer)
6. @i16 (16 bit Integer)
7. @d (64 bit float)
8. @u (Make any numeric data type unsigned)
9. if
10. else
11. check
12. case
13. stop
14. continue
15. default
16. for
17. while
18. do
19. forall
20. use
21. give
22. test
23. fail
24. last
25. class
26. static
27. open
28. secret
29. permitted
30. declared
31. implicit
32. recall
33. include
34. closed
35. enums
36. @i:

Used to declare integer variable.

Syntax:

@i [variable\_name] = value;

1. @f:

Used to declare floating point value.

Syntax:

@f [variable\_name]=value;

1. @c:

Used to declare character variable.

Syntax:

@c [variable\_name] = ‘character’;

1. @b:

Used to declare Boolean variable.

Syntax:

@b [variable\_name] = True/False;

1. @i64:

Used to declare integer variable having 64 bit value.

Syntax:

@i64 [variable\_name] = value;

1. @i16:

Used to declare integer variable having 16 bit value.

Syntax:

@i16 [variable\_name] = value;

1. @d:

Used to declare floating point variable having 64 bit value.

Syntax:

@d [variable\_name] = value;

1. @u:

It can be placed prior to any numeric data type so that it can make that data type value range unsigned i.e. positive values.

Syntax:

@ui [variable\_name] = value;

@uf [variable\_name] = value;

@ui64 [variable\_name] = value;

@ui16 [variable\_name] = value;

@ud [variable\_name] = value;

1. if:

if keyword is used to put any condition on any code block for execution.

Syntax:

if (Any conditional statement)

Statement to be executed;

OR

if (Any conditional statement ){

Statements to be executed;

}

1. else:

else keyword comes after if to execute any other alternate statements if *if* condition is false

Syntax:

if (Any conditional statement){

Statements to be executed;

}

else{

Alternate statement to be executed;

}

1. check:

check keyword is another kind of control statement commonly known as switch case in other languages.

Syntax:

check(value){ //*check any value passed as an argument*

case value1: *//if passed value is value1*

execute this statement; //run this statement

stop; //break out of the switch block

case value2: // if passed value is value2

execute this statement; //run this statement

stop; // break out of the switch block

default: // if neither case is true

execute this statement; // run this code of block

stop; // break out of the switch block

}

1. continue:

continue key word is used to pass the control over next iteration in loop execution.

Syntax:

continue;

1. for:

for keyword is used to run a set of commands again and again until the given condition is false.

Syntax:

for([variable\_name] = value; [condition] ; iterator){

Statements to be executed;

}

1. while:

while keyword is used to execute a block of statement until the specified condition is met.

Syntax:

while( condition ){

statement to be executed;

}

1. do:

do keyword is used with while as a do-while loop.

Syntax:

do{

Statements to be executed;

}

while(condition);

1. forall:

forall keyword is used to run loop which will go through every item of a collection.

Syntax:

forall([variable\_name] in [collection]){

statements to be executed;

}

1. use:

use keyword is used to declare function.

Syntax:

use [return\_type] [function\_name]([parameters]){

statements to be executed;

}

1. given:

given keyword is used to return any value from the function.

Syntax:

given [return\_type];

1. test/fail:

test keyword is used to declare a block that will handle exception followed by fail block.

Syntax:

test{

Statements to be executed;

}

fail([exception]){

Statements to be executed;

}

1. last:

last keyword is used to declare a block that will run when both test and fail case fail to run.

Syntax:

test{

statements to be executed;

}

fail([exception]){

statements to be executed;

}

last{

statements to be executed;

}