Cairo University

Compilers Project Document

Team :16

Name	Section	B.N
Aly Ramzy Hassan	1	35
Mohamed Ahmed Ibrahim	2	9
Nour Ahmed	2	29
Hager Ahmed	2	32
Ali Khaled	1	34

Supported data types

1. int : to represent integer numbers.
2. float : to represent decimal numbers, or numbers with exponent part.
3. char : to represent a single character.
4. bool : true/false.
5. void : for functions as a return type.

Syntax

1. Variables and constants declaration:

$To\ define\ a\ constant$:

$const\ dataType\ identifier\ =\ value$;

$Example$: $const\ int\ x\ =\ 5$;

$To\ define\ a\ variable$:

$dataType\ identifier\ =\ value$;

$Example$: $int\ x\ =\ 5$;

2. Mathematical and logical expressions.

Supported operations :

- Addition , Subtraction , Multiplication , Division , Modulus .
- & , | , ^ , || , && .
- < , > , >= , <= .
- ~ , ! .
- Any level of pranthess/complexity .

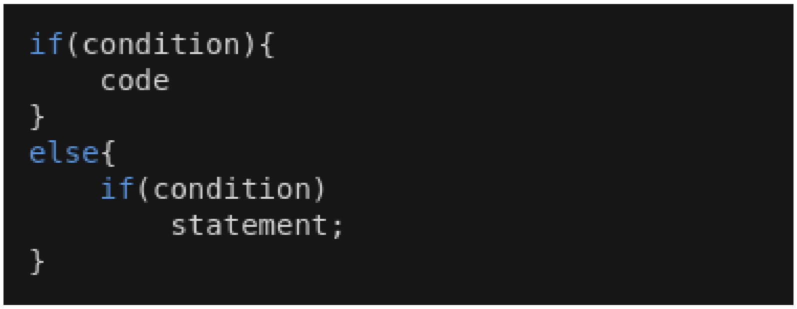
3. Assignment statement

$variable\ =\ expression$;

$Example$: $x\ =\ 5$;

4. If-then-else statement

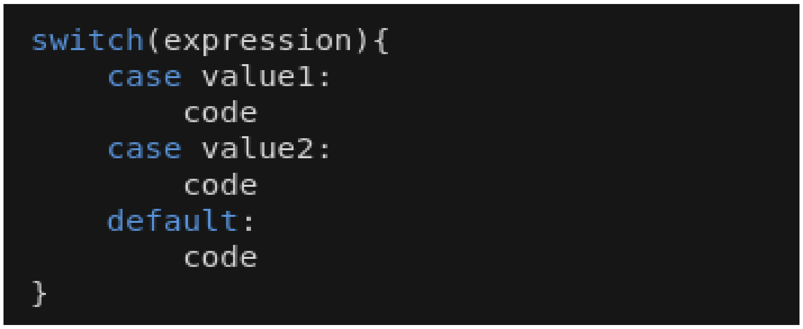
Supports conditional statements like C language.



```
if(condition){
    code
}
else{
    if(condition)
        statement;
}
```

5. Switch statement

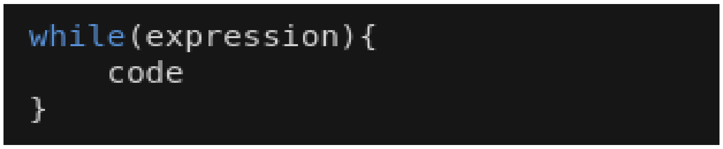
C-like.



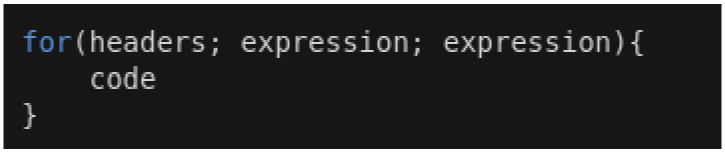
```
switch(expression){
    case value1:
        code
    case value2:
        code
    default:
        code
}
```

6. Loops

- While/for loops are C-like.

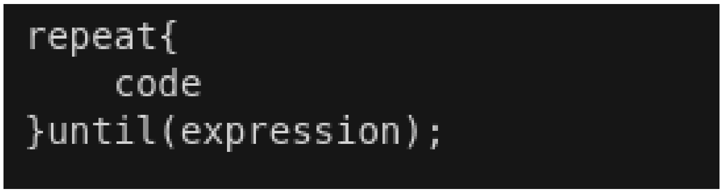


```
while(expression){
    code
}
```



```
for(headers; expression; expression){
    code
}
```

- Repeat-until has the following syntax:

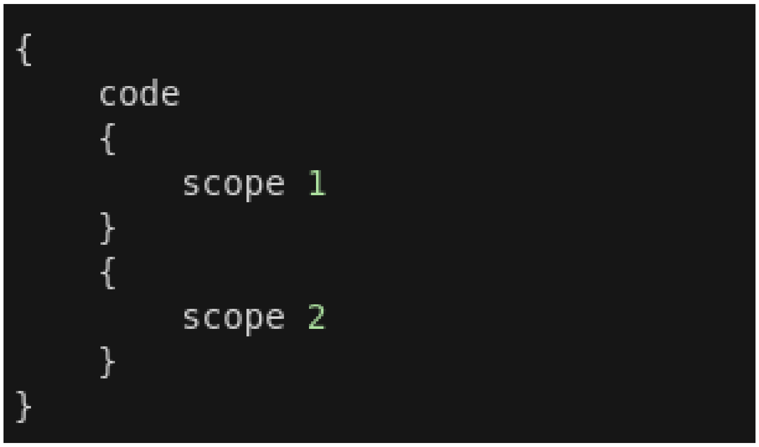


```
repeat{
    code
}until(expression);
```

7. Block structures

Each block represents a scope.

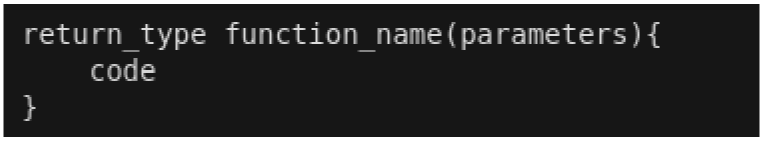
Variables within a scope are not visible to other scopes, excluding some semantic cases that would be determined later.



```
{
    code
    {
        scope 1
    }
    {
        scope 2
    }
}
```

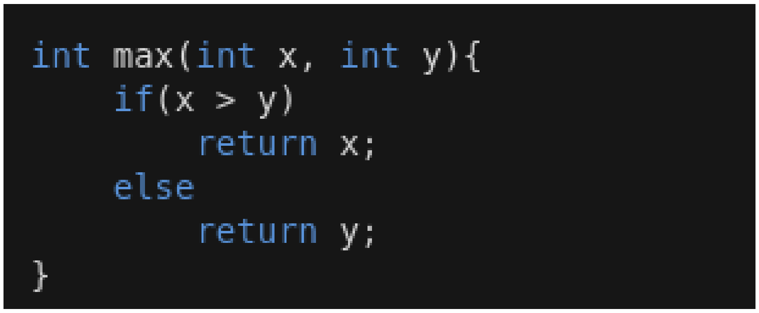
8. Functions

Functions have the following syntax:



```
return_type function_name(parameters){
    code
}
```

For example:



```
int max(int x, int y){
    if(x > y)
        return x;
    else
        return y;
}
```