

- [Team 20](#)
- [How to run](#)
 - [To run the test.txt file](#)
 - [To run a custom file](#)
- [Language Descriptions](#)
 - [Tokens](#)
 - [1- Variable and constant Declaration](#)
 - [2- Mathematical and Logical expressions](#)
 - [3 - Assignment statement](#)
 - [4 - If else statement](#)
 - [5 - While loop](#)
 - [6 - Repeat until](#)
 - [7 - For loop](#)
 - [8 - Switch case](#)
 - [9 - Function declaration](#)
 - [10 - Function call](#)
 - [11 - Block structure](#)
 - [14- Enum](#)
 - [15 - Type declaration \(optional\)](#)

Team 20

Osama Magdy 1 14

Reem Emad 1 33

Kareem Taha 2 9

Yousef Gamal 2 39

How to run

To run the test.txt file

1. `chmod u+x ./test.sh`
2. `./test.sh`

To run a custom file

1. make
2. ./comp <yourfile>

Language Descriptions

Tokens

1- Variable and constant Declaration

```
val x: Int = 10;

var x: Int = 10;

val x: Int;
var x: Int;
```

- - int
 - double
 - string
 - bool
 - void

2- Mathematical and Logical expressions

- Mathematical operator
 - +, -, *, /, **, %
- Logical operator
 - <, >, <=, >=, ==, !=
 - and , or , not ,

3 - Assignment statement

```
x = 5;  
x = "Hello";  
x = 5.5;  
x = true;
```

4 - If else statement

```
if (x < 5 and y > 10) {  
    x = x + 1;  
} else {  
};
```

```
if 1 {  
    x = x +1 ;  
} else {  
    if (x < 5 and y > 10) {  
        x = x + 2;  
    } else {  
        x = x + 3  
    };  
};
```

```
if (x < 5 and y > 10) {  
    x = x + 2;  
} else {  
    x = x + 3  
};
```

5 - While loop

```
while (1) {  
    x = x + 1;  
};
```

6 - Repeat until

```
until (x == 5){  
    x = x + 1;  
};
```

7 - For loop

```
for i:(x,2,4) {  
    x = x + 1;  
};
```

8 - Switch case

```
switch x {  
    1:{  
  
    }  
    1:{  
  
    }  
    2:{  
  
    }  
    else:{  
  
    }  
};
```

9 - Function declaration

```
fun func1(int x, int y) int {  
    var z: Int = x + y;
```

```
z;  
}
```

```
fun func2(int x) void {  
    x = x + 1;  
}
```

```
fun func3() void {  
    x = x + 1;  
}
```

10 - Function call

```
x = func1(5, 10);  
func2(5, 10,  
566,  
568,);  
func3();
```

11 - Block structure

```
do {  
    do{  
        do{  
            };  
        };  
    };  
};
```

14- Enum

```
type BOOL = TRUE | FALSE | OTHER;
```

15 - Type declaration (optional)

```
type Bool = MkInt {str:int, kak:KKK}  
          | MkBool{str:int, kak:KKK}  
          ;
```