Compilers & Languages Project

Team 3

Name	ID	SEC	BN
Asmaa Adel Abdelhamed kawashty	9202285	1	13
Samaa Hazem Mohamed Abdel-latif	9202660	1	31
Norhan Reda Abdelwahed Ahmed	9203639	2	31
Hoda Gamal Hamouda Ismail	9203673	2	33

Lexer :

Keywords:

- int, float, double, string, enum, bool
- print, void, return, continue
- switch, break
- case, default
- if, else
- for, while, do
- true, false, and, or

Operators:

- "==", "!=", "--", "++", "!" , "~"
- ">", "<", "="
- "%", "+", "-", "*", "/", "^"
- "<<", ">>"

Constants:

- [A-Z]+

Identifier:

- [a-zA-Z_][a-zA-Z0-9_]*

Parser:

Print:

- print("lol");

Variables declaration:

- int a = 5; float y=3.4;

Constant declaration:

- Capital letters only
    - Example: int X=6;

Mathematical operations:

- +, -, *, /, %, ^, ++, –
    - x= x+1 ;

Logical operations:

- and, or, !
    - if (x and y){...}

Control expressions:

- "==", "!=", ">", "<", ">=", "<="

Bitwise operator:

- "~"

Shift operator:

- "<<", ">>"

Assignment statements

- Example: a = 5;

If-then-else statement:

```
if x > 0 {
    print("x is positive");
  } else {
```

```
        print("x is non-positive");
  }
```

For Loop:

```
for (i = 0; i < 10; i = i + 1;) {
    print(i);
}
```

While Loop:

```
while x > 0 {
    x = x - 1;
}
```

Do-While Loop:

```
do {
    x = x - 1;
} while (x == 0);
```

Switch Statements

```
switch x: {
    case 0: print("x is zero"); break;
    case 1: print("x is one"); break;
    default: print("x is neither zero nor one");
}
```

Block structure:

```
{
    int x;
    {
        int y;
    }
}
```

Functions:

```
int square(int num) {
    return num * num;
}
result = square(5);
```

Comments:

- Start line with ‘#’
    - Example: # this is a comment

Enums:

```
enum test {id1, id2, id3}
```

How to run:

Using terminal:

- flex lexer.l
- bison -d parser.y
- gcc *.c -o out.exe
- write your Code inside a file, for example in.txt
- .\out.exe in.txt

Using GUI:

- Open CMD in the GUI directory, type python ctk.py
- Write your Code inside
- Press Compile!

Tokens:

DIGIT	[0-9]+
FLOAT	[0-9]+\.[0-9]+
IDENTIFIER	[a-zA-Z_][a-zA-Z0-9_]*
STRING	\"[a-zA-Z0-9_]+\"
CONSTANT	[A-Z]+
DATA TYPES	int, float, double, string, enum, bool
OTHERS TOKENS	print, void, return, switch, break, continue, case default, if, else, for, while, do
SPACE	[\t]+
NEW LINE	\n
COMMENT	#[\s\t]*.*[\s\t]*
ARITHMETIC Ops	"%", "+", "-", "*", "/", "^","--", "++"
SHIFT Ops	"<<", ">>"
BITWISE Ops	"~"
CONTROL Ops	"==", "!=", ">", "<", ">=", "<="
SEMICOLON	";"
true	true
false	false
PUNCTUATION	){}(:,]