

# SCS3211 Compiler Theory

## Assignment 1

**Goal** : Acquaint yourself with flex and bison.

Done by a group of 2 students.

You will extend calc.l and calc.y to parse and type check programs whose syntax is defined below.

**Prog**  $\rightarrow$  **main()** {**Stmts**}

**Stmts**  $\rightarrow$   $\epsilon$  | **Stmt**; **Stmts**

**Stmt**  $\rightarrow$  **int** **Id** | **float** **Id** | **Id** = **E** | **printvar** **Id**

**E**  $\rightarrow$  **Integer** | **Float** | **Id** | **E** + **E** | **E** \* **E**

**Integer**  $\rightarrow$  **digit**<sup>+</sup>

**Float**  $\rightarrow$  **Integer** . **Integer**

**Prog**: a program

**Stmts**: empty or a sequence of statements separated using ;

- **Id**: a sequence of one or more lower-case letters or digits (start with lower-case letters).
  - E.g. x1, xy are identifiers, but 1x and A are not.
- Each variable is either a positive integer (int) or a positive floating point (float).

**Expression E**

- An integer
- An identifier
- An infix arithmetic expression
- + and \* are left associative
- has higher precedence than +

**Id = E** assigns the value of an expression E to Id **printvar Id** outputs the value of the expression Id.

Tokens may be separated by any number of **white spaces**, **tabs** or **new lines**.

If there is a **parse error**, you need to report the error.

- Your error messages must contain the line number where the error was found.

### Type Checking

`Stmt → int Id | {id.type = 0}`

`float Id | {id.type = 1}`

`Id = E {if (Id.type != E.type) then type error}`

`E → Integer | {E.type = 0}`

`Float | {E.type = 1}`

`Id | {E.type = Id.type}`

`E1 + E2 | {if (E1.type==E2.type) then E.type = E1.type;  
else type error}`

`E1 * E2 | {if (E1.type==E2.type) then E.type = E1.type;  
else type error}`

### Compile your code

`flex -l calc.l`

`bison -dv calc.y`

`gcc -o calc calc.tab.c lex.yy.c -lfl`

### Execution (example):.

`./calc < input`

Where input is the name of the input file

### Example testcases

Program 1:

`main() {int x; x = 3;}`

Output:

Program 2:

`main() {int x; x = 3; printvar x;}`

Output:

Program 3:

```
main() {int x; x = 3; x = x + 4; printvar x;}
```

Output:

Program 4:

```
main() {x=3;}
```

Output:

Program 5:

```
main() {int x; x = 3;}
```

Output:

Program 6:

```
main() {int x; x = 3; printvar x;}
```

Output: 3

Program 7:

```
main() {int x; x = 3; x = x + 4; printvar x;}
```

Output:

Program 8:

```
main() {x=3;}
```

Output:

Program 9:

```
main() {int x; x = 3;}
```

Output:

Program 10:

```
main() {int x; x = 3; printvar x;}
```

Output: 3

Program 11:

```
main() {int x; x = 3; x = x + 4; printvar x;}
```

Output: 7

Program 12:

```
main() {x=3;}
```

Output: Line 1: x is used but is not declared

Program 13:

```
main() {int x; x = 1-2; }
```

Output:

Program 14:

```
main() {  
int1x;  
}
```

Output:

Program 15:

```
main() {  
    int x;  
    x = 3;  
    print x;  
}
```

Output:

### Submission Guidelines

Hand in your source code and a makefile electronically (do not submit .o or executable code).

Each group uploads only ONE copy of the assignment

Make sure that this code compiles and runs correctly on linux. The makefile must give the executable code the name calc

Write a README file (text file, do not submit a .doc file) which contains

- Names, section numbers, and email addresses of group members
- Whether your code was tested on linux.
- How to execute your program.
- Briefly describe anything special about your submission that the instructor should take note of.

Place all your files under one directory with a unique name (p1-[fname1\_fname2] for assignment 1, e.g. p1- yang\_fang).

Tar the contents of this directory using the command

```
tar -cvf [directory_name].tar [directory_name]
```

E.g. tar -cvf p1-yang.tar p1-yang/

### **Grading Guidelines**

README, Correct executable names: 4'

Correct makefile: 8'

Lexical analysis, parsing, output: 64'

Type checking: 24'

### **Academic Integrity**

We will use moss to detect to plagiarism in this assignment.

Use the LMS to upload the tared file you created above.