

# Μεταγλωττιστές 2023-24

Προσθήκη συναρτήσεων στη γραμματική των  
αριθμητικών εκφράσεων

# Στόχος

- Η υποστήριξη απλών συναρτήσεων από τον συντακτικό αναλυτή και AST interpreter
  - Δηλώνονται στην αρχή του προγράμματος
    - `function func-name (argument-list) { statements }`
  - Καλούνται είτε ως statement είτε ως expression
    - `func-name (expression-list)`
  - Μπορούν να επιστρέφουν μια float τιμή
    - `return expression`
    - Για απλοποίηση, το return θα επιτρέπεται ακόμα και στο «κυρίως πρόγραμμα» που ακολουθεί μετά τις δηλώσεις των συναρτήσεων

# ΣΥΝΤΑΚΤΙΚΗ ΑΝΑΛΥΣΗ

- Προσθέστε στη γραμματική τους νέους κανόνες
  - Γραμματική για να αρχίσετε με το on-line εργαλείο στο <http://smlweb.cpsc.ucalgary.ca/start.html>

```
Stmt_list -> Stmt Stmt_list | .
Stmt -> id eq Expr | print Expr
      | if Expr Block_stmt | while Expr Block_stmt .
Block_stmt -> Stmt | lbr Stmt_list rbr .
Expr -> Term Term_tail .
Term_tail -> Addop Term Term_tail | .
Term -> Factor Factor_tail .
Factor_tail -> Multop Factor Factor_tail | .
Factor -> lpar Expr rpar | id | num .
Addop -> plus | minus .
Multop -> mult | div .
```

## ΣΥΝΤΑΚΤΙΚΗ ανάλυση (2)

- Δηλώστε τα νέα patterns του tokenizer
  - Keywords και σημεία στίξης
- Υλοποιήστε τις μεθόδους των νέων μη τερματικών συμβόλων
  - Και τροποποιήστε αν χρειαστεί τις υπάρχουσες μεθόδους
    - Αν έχουν αλλάξει τα FIRST/FOLLOW sets

# ΣΥΝΤΑΚΤΙΚΗ ανάλυση (3)

- Δοκιμάστε τη λειτουργία του συντακτικού αναλυτή με τη διπλανή δοκιμαστική είσοδο
  - Χωρίς κατασκευή και διερμηνεία AST

```
function um(x)
  return 0-x
function cube(x) {
  return x*x*x
}
a = 2 + 7.55*44
print a
if a-7 {
  b = 3*(a-99.01)
  it = 5
  while it {
    print it+b*0.23
    it = it - 1
  }
}
c = 5-3-2
if it+1 if c print c else print
cube(um(c+28))
```