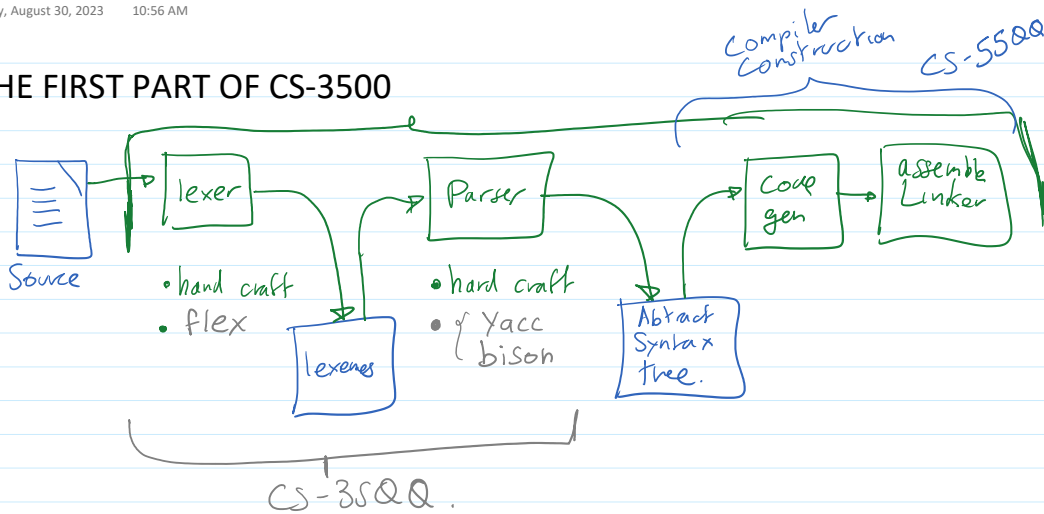


• THE FIRST PART OF CS-3500



• THE PUCK 23.3 PROGRAMMING LANGUAGE

Imperative in style  
C, Java, Python.

Basic Components {  
 - assignment - expressions.  
 - conditionals - comparison.  
 - loops. - functions

```
% This is a comment
PROCEDURE main ( )
  WRITE ( "Hello" , "World" ) ;
  x := 2 + 2 ;
  y := 3 * 12 / 7.5 ;
  p := ( x > 0 ) AND ~ ( y <= 30 ) ;
  WRITE ( x * 100 ) ;
END.
```

- Keywords are uppercase.
- := assignment ALGOL'60
- strings are separated by spaces.
- Literals: Numbers, strings.
- relational as usual

AND  
 ; is a separator, not a terminator.

```
% FizzBuzz
PROCEDURE FizzBuzz ( n )
  IF n MOD 3 = 0 THEN
    IF n MOD 5 # 0 THEN
      WRITE ( "Fizz" )
    ELSE
      WRITE ( "FizzBuzz" )
    FI
  ELSE
    IF n MOD 5 = 0 THEN
      WRITE ( "Buzz" )
    FI
  FI
END.
```

MOD # MOD // DIV  
 bye bye { }  
 #

% Functions and Loops

```
FUNCTION fibo ( n )
```

```
  x := 1 ;
```

```
  y := 2 ;
```

```
  c := 3 ;
```

```
  LOOP c < n DO
```

```
    x := x + y ;
```

```
    y := x - y ;
```

```
    c := c + 1 ;
```

```
  POOL
```

```
  RETURN (x) END.
```

PROCEDURE

don't return

VS

FUNCTION

return

no semi\_colon here.

Return is part of the function

% Greatest Common Denominator

```
FUNCTION gcd ( a , b )
```

```
  LOOP a # b DO
```

```
    IF a > b THEN
```

```
      a := a MOD b
```

```
    ELSE
```

```
      b := b MOD a
```

```
    FI
```

```
  POOL
```

```
  RETURN a END.
```

& string concatenation.

% Greatest Common Denominator

```
PROCEDURE hello ( ) WRITELN (
```

```
  "Hello"
```

```
  &
```

```
  "World"
```

```
  ) END.
```

in Math

$$a + b \equiv b + a$$

% Newlines are just whitespace

```
FUNCTION foo ( s )
```

```
  res := false ;
```

```
  IF ( s = "<" ) OR ( s = ">" ) OR
```

```
    ( s = "=" ) OR
```

```
    ( s = "#" ) THEN
```

```
    res := true
```

```
  FI
```

```
  RETURN res END.
```

% IF and LOOP may need semicolons

```
FUNCTION zap ( s )
```

```
  res := 3 ;
```

```
  IF < > 0 THEN
```

```

FUNCTION zap ( s )
  res := 3 ;
  IF s > 0 THEN
    WRITE ( "zero" )
  FI ;
  LOOP s > 0 DO
    s := s - 1
  POOL ;
  WRITELN ( "Done!" )
RETURN res END.

```

Python 2.7 ~~→~~ 3.0

C++ 03 ↔ 07 ↔ 11 ↔ 14 ↔ 17 ↔ 20 ↔ 23



-E.O.F.-