

3 "Bench" Programming Language

Monday, February 3, 2025 12:28 PM

Basic Components

- literals.
- operators
- Expressions.
- Variables.
- Assignment.
- Conditional
- Repetition
- "Functions"

The PUCK 25.1 Programming Language:

```
% 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 main .
```

• keywords are UPPERCASE

• := assignment.

• Space separation.

• Literals

• Operators

~ negation.

• ; is a "separator," not a "terminator"

• name of function repeats.

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

• # not equal

• MOD modulus

• DIV integer division.

• ELIF

% Fibonacci and Loops

```
PROCEDURE fibo ( n )  
  x := 1 ; y := 2 ; c := 3 ;  
  WHILE c < n DO  
    x := x + y ;  
    y := x - y ;  
    c := c + 1  
  END  
  RETURN x END fibo .
```

- Indentation not important
- WHILE
- RETURN is part of END.

% Greatest Common Denominator

```
PROCEDURE gcd ( a , b )
```

```
  WHILE a > b DO  
    a := a MOD b  
  ELSIF b > a DO  
    b := b MOD a  
  END
```

```
  //  $\neg(a > b) \text{ AND } \neg(b > a) \equiv a = b$   
  RETURN a END gcd .
```

- WHILE has an ELSIF,
- The conditions are tested sequentially until one of them is true.
 - the corresponding block is executed
 - the next iteration continues.

% Newlines are just Whitespace

```
PROCEDURE foo ( s )
```

```
  res := True
```

```
  IF ( s = "A" ) OR  
    ( s = "B" )
```

```
  THEN
```

```
    WRITE
```

```
    (
```

```
      "Hello" & "World"
```

```
    )
```

```
  END END foo .
```

$$a + b \neq b + a$$

String
concatenation.

—•— EOF