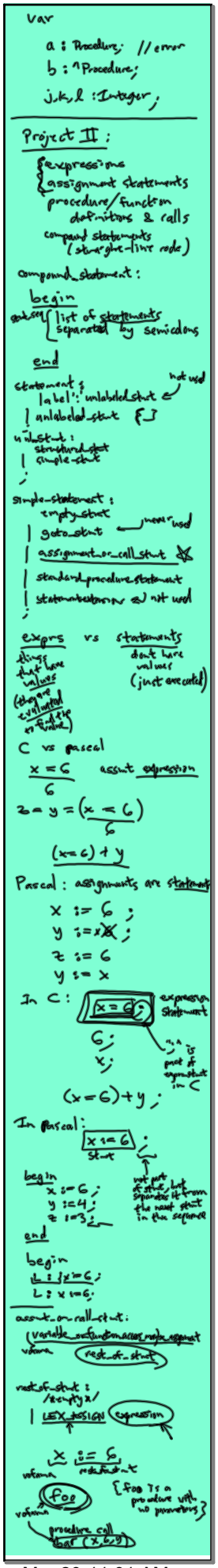


Mar 20-11:01 AM

```
Function Foo : Integer;
   }
Procedure Bar ;
```

In Pascal, procedure calls are statements;
function calls are expressions

```
x := Foo    // ok
Bar         // ok
x := Bar    // semantic error

Function Bat : IntPtr;

Bat^ := 6    // okay!
```

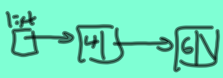
```
vofana:
    identifier
  | address_operator variable_or_function_access   <- not used

  | variable_or_function_access_no_id
          "vofani"
;
vofani:
    :
  | '(' expression ')'
  | var_or_fcn_access pointer_char
      some expr
      of pointer type
  | var_or_fcn_access '[' index_expr_list ']'
  | var_or_fcn_access_no_standard_fcn '(' actual
                                   param_list ')'
  | pNEW '(' var_or_access or typename ')'
;
var_or_fcn_access:
    var_or_fcn_access_no_as
  | var_or_fcn_access LEX_AS typename   <- not used
;

var_or_fcn_access_no_as:
    var_or_fcn_access_no_std_fcn
  | standard_functions
;
var_or_fcn_access_no_standard_fcn:
    identifier
  | vofani
```

```
type
   ListNode = record
                 data : Integer;
                 next : ^ListNode;
              end;

var
   list : ^ListNode;
   p : ^ListNode;
   :
begin
   new(list);
   list^.data := 6;
   list^.next := nil;
   new(p);
   p^.data := 4;
   p^.next := list;
   list := p;
```



Mar 20-11:49 AM