

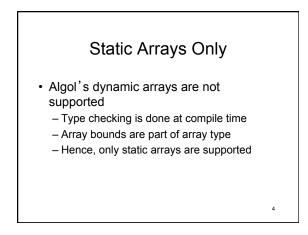
Dimensions

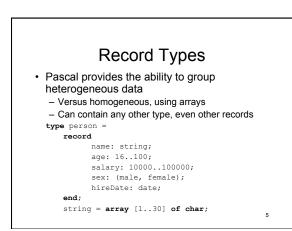
- Only single-dimension arrays are allowed!!!
- However:

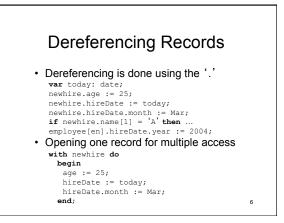
 Base type of array can be another array!!!
 var M: array [1..20] of array [1..100] of real;
 Dereferencing: M[3] [5]

3

• Syntactic sugar. var M: array [1..20, 1..100] of real; M[3, 5] (Doesn't affect functionality, sweeter for human use.)







Variant Records

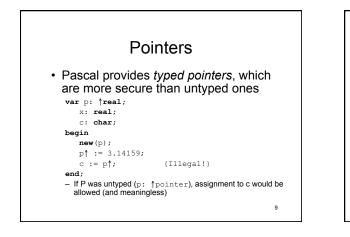
- Pascal supports saving storage using variant records; allows alternative structures
 - Not all components of a record may be used at the same time

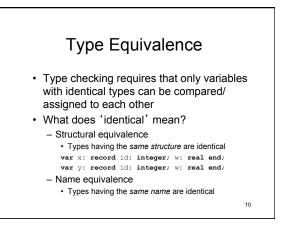
E.g.: Plane altitude and location on ground

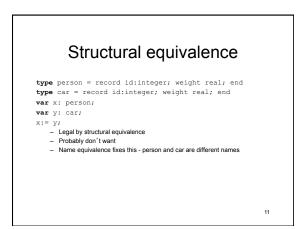
- C: union
- Union is unsafe as it allows access to any member
 Pascal attempts to solve this security problem
 - Access only members allowed by tag field
 - Initialization not required after tag value change, so type system can be circumvented after all...

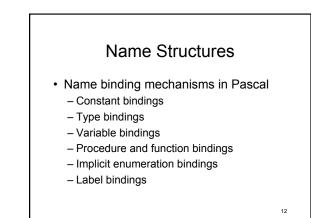
Variant Record Example

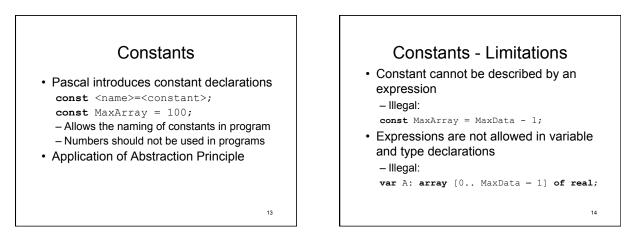


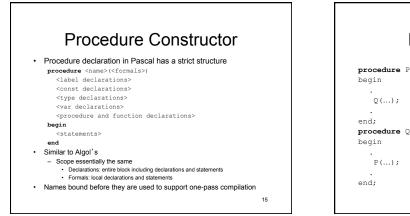


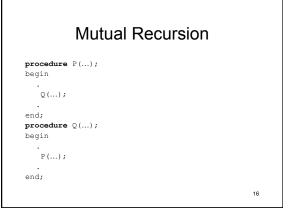








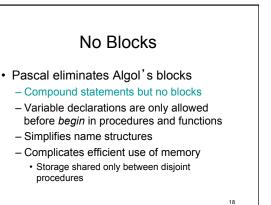




Procedure Constructor

- Opposite of top-down
 - Uppermost procedures first, then lower ones they call
- Mutual recursion
 - Cannot define both procedures before one is called
- · Pascal's solution
 - "forward" declaration of procedures allows recursion, and observation of structure principle procedure Q(...); forward;

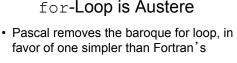
17



3

Control Structures

- Pascal includes more control structures than Algol-60, but they are simpler
 - Provides simple I/O
 - Introduces more structured control structures (structure principle)
 1-entry point 1-exit point controls
 - Includes goto (rarely needed)
 - Includes gete (rerely needed)
 Includes recursive procedures



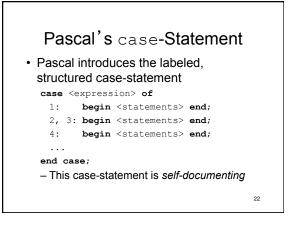
for <name> := <exp> {to|downto} <exp> do
 <statement>

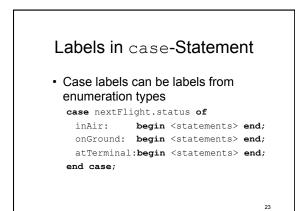
- Only step size of 1 is allowed (+1 & -1)
 May be too restrictive
- Bounds are computed once, on entry
 - Called *definite iterator*
 - Always executes a definite number of times unless goto

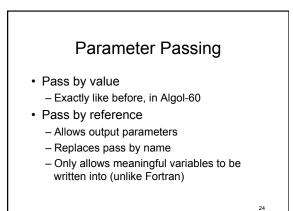
Leading & Trailing Decision Loops

19

- Indefinite iterators:
 - Loop is controlled by condition, not counter
 Condition is tested each time
- Versus pre-computed in *for*-loop
 Leading Decision loop
- while <condition> do <statement>
 Trailing Decision loop
- repeat <statement>* until <condition>
 Mid-Decision loop
- Can be implemented using "while true do" and goto 21







26

Pass as Constant

- Pass as constant was originally specified instead of pass by value
 - Like pass by value, but parameter could not be modified in callee
 Safe
 - Implemented as pass by reference
 Efficient
 - Replace by pass by value, since pass as constant can be circumvented using scoping (p 202)
 C++ provides this functionality by explicit pass by reference and const definitions (f(const int &a))

Two Orthogonal Issues

- · Input vs output parameters
- Copy value vs pass address
- · Decisions should be separated

Goals

- · Main goal: good teaching language
 - Reliability
 - Simplicity
 - Efficiency
- Successful!
- Third Generation

27