Arrays

In this section of notes you will be introduced to a homogeneous composite type, one-dimensional arrays

Simple Types (Atomic)

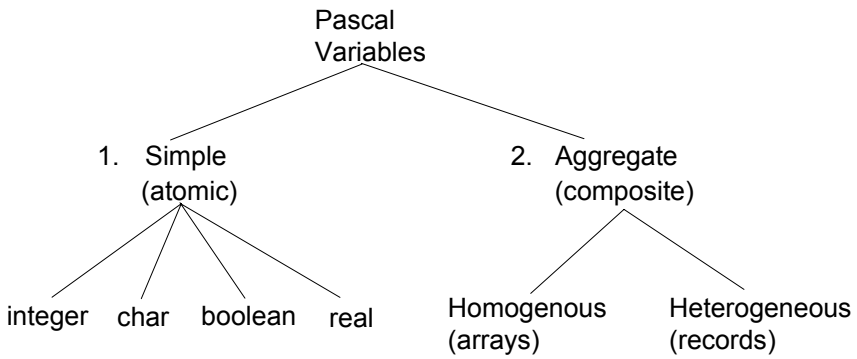
1) Integer
2) Real
3) Char
4) Boolean

Composite Types (Aggregate)

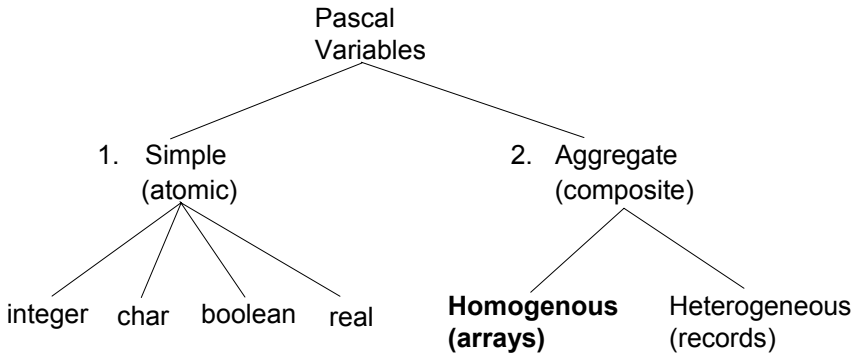
1) ***Homogeneous***
   - ***arrays***

2) Heterogeneous
   - records

Types Of Variables



Types Of Variables



Why Bother With Composite Types?

For a compilable example look in Unix under:
/home/231/examples/arrays/classList1.p

```
const

   CLASSSIZE =  5;

begin

   var stu1, stu2, stu3, stu4, stu5, total, average : real;

   write('Enter grade for student number 1: ');

   readln(stu1);

   write('Enter grade for student number 2: ');

   readln(stu2);
```

Why Bother With Composite Types (2) ?

```
write('Enter grade for student number 3: ');
readln(stu3);
write('Enter grade for student number 4: ');
readln(stu4);
write('Enter grade for student number 5: ');
readln(stu5);
total := stu1 + stu2 + stu3 + stu4 + stu5;
average := total / CLASSSIZE;
writeln('The average grade is ', average:6:2, '%');
```


With Bother With Composite Types (3)

```
(* Printing the grades for the class. *)
writeln('Student: ', 1, stu1);
writeln('Student: ', 2, stu2);
writeln('Student: ', 3, stu3);
writeln('Student: ', 4, stu4);
writeln('Student: ', 5, stu5);
```

With Bother With Composite Types (3)

```
(* Printing the grades for the class. *)
writeln('Student: ', 1, stu1);
writeln('Student: ', 2, stu2);
writeln('Student: ', 3, stu3);
writeln('Student: ', 4, stu4);
writeln('Student: ', 5, stu5);
```

NO!

Revised Version Using An Array

For a compilable example look in Unix under:
/home/231/examples/arrays/classList2.p

```
const
  CLASSSIZE =  5;
begin
  var classGrades    : array [1..CLASSSIZE] of real;
  var i              :  integer;
  var total, average : real;
  total := 0;
```

Class Example Using An Array (2)

```
for i := 1 to CLASSSIZE do
begin
    write('Enter grade for student no. ', i, ': ');
    readln (classGrades[i]);
    total := total + classGrades[i];
end;
average := total / CLASSSIZE;
writeln;
writeln('The average grade is ', average:6:2, '%');

for i := 1 to CLASSSIZE do
    writeln('Grade for student no. ', i, ' is ', classGrades[i]:6:2, '%');
```

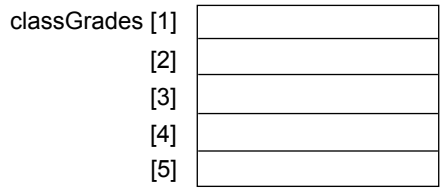

Declaring Arrays

Format:

Name: array [*low index..high index*] of *element type*;

Example:

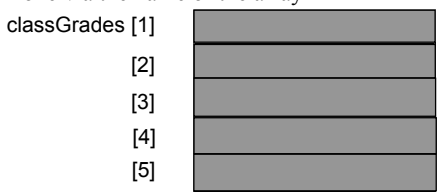
classGrades : array [1..CLASSSIZE] of real;



Accessing Data In The Array

First you need to indicate which array is being accessed

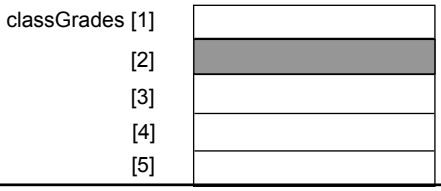
- Done via the name of the array



I

f you are accessing a single element, you need to indicate which element that you wish to access.

- Done via the array index



James Tam

Accessing Data In The Array (2)

Format:

(Whole array)	(One element)
name	name [index]

Examples (assignment via the assignment operator):

(Whole array)	(One element)
firstArray := secondArray;	classGrades [1] := 100;

James Tam

Accessing Data In The Array (3)

Examples (assigning values via read or readln):

(Single element)

```
readln(classGrades[1]);
```

(Whole array – all elements)

```
for i: = 1 to CLASSIZE do
begin
    write('Input grade for student No. ', i, ': ');
    readln(classGrades[i]);
end;
```


Accessing Data In The Array (4)

(Whole array – all elements: Character arrays only)

```
var charArray : array [1..5] of char;
readln(charArray);
```

Accessing Data In The Array (5)

Examples (displaying information):

(Single element)

```
writeln(classGrades[1]);
```

(Whole array – all elements)

```
for i := 1 to CLASSSIZE do
    writeln('Grade for student No. ', i, ' ', classGrades[i]);
```

(Whole array – all elements: Character arrays only)

```
var charArray : array [1..5] of char;
write(charArray);
```


Common[1] Array Operations

Declaration
- Done previously in this set of notes (slide No. 12)

Assignment
- Done previously in this set of notes (slide No. 9, loop body line No. 2)

Extracting Elements
- Single element – done previously in this set of notes (slides No. 15)
- All elements – done previously in this set of notes (slide No. 16)

In order copy between two arrays
- Using the assignment operator – done previously in this set of notes (slide No. 14)
- *Manual copy – coming up*

Reverse order copy between two arrays
- *Manual copy – coming up*

1) Common but by no means complete

In-Order Copy Between Arrays

Method 1: Using the assignment operator
- e.g., array1 := array2;

Method 2: Manual copy
- Use loops and copy from one array to another element-by-element

Example of manual copy (full example can be found in Unix under /home/231/examples/inOrderArrayCopy.p)

```
const
  SIZE      =  5;
  MAXVALUE  = 11;
begin
  var array1 : array [1..SIZE] of integer;
  var array2 : array [1..SIZE] of integer;
  var i      : integer;
```


In-Order Copy Between Arrays (2)

```
for i:= 1 to SIZE do
  array1[i] := RANDOM (10);

for i:= 1 to SIZE do
  array2[i] := array1[i];

writeln;
for i:= 1 to SIZE do
   writeln('array1: ', array1[i]:2, '     array2: ', array2[i]:2);
```

Reverse Order Copy Between Arrays

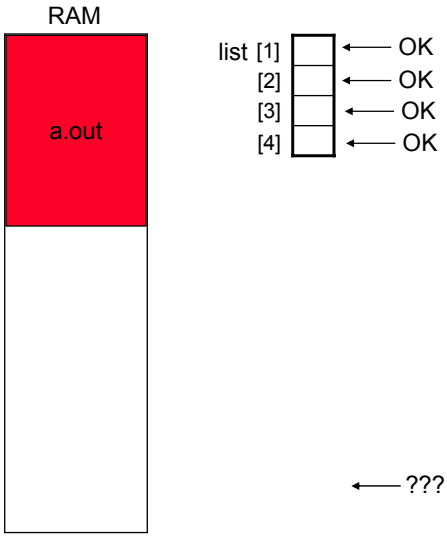
Example of reverse order copy (full example can be found in Unix under /home/231/examples/reverseOrderArrayCopy.p)

```
program reverseOrderArrayCopy (output);

const

  SIZE = 5;

begin

 var array1 : array [1..SIZE] of integer;

 var array2 : array [1..SIZE] of integer;

  var i, j     : integer;

  for i:= 1 to SIZE do

    array1[i] := RANDOM(10);
```


Reverse Order Copy Between Arrays (2)

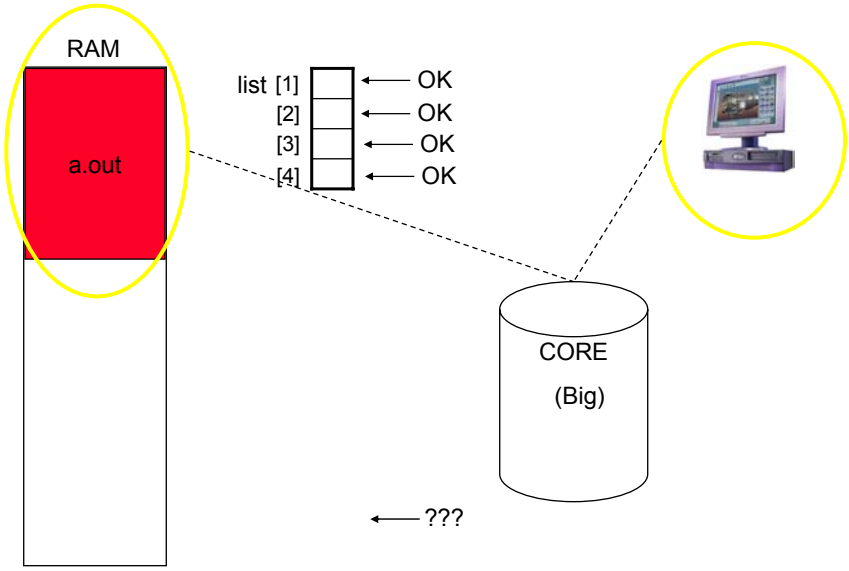
```
  j := SIZE;

 for i:= 1 to SIZE do

  begin

    array2[j] := array1[i];

    j := j - 1;

  end;

for i:= 1 to SIZE do

    writeln('array1: ', array1[i]:2, '     array2: ', array2[i]:2);

  writeln;

end.
```

Segmentation Faults And Arrays



James Tam

Segmentation Faults And Arrays



Wav file from “The SImpsons”

James Tam

You Should Now Know

What is the difference between simple types (atomic) and composite types (aggregate)

What is the benefit of using homogeneous composite types (arrays)

How to declare arrays

How to access or assign values to array elements

How to work with an entire array

How to manipulate arrays in different ways, such as through the common array operations