

Examples of Type Unions

- **Examples of Pascal variant records**
 - Storing employee data for different kinds of employees
 - Using integer divide and multiply to simulate bit shifts

Pascal variant record

```
program main(input,output);
type paytype = (salaried,hourly);
var employee : record
  id : integer;
  dept: integer;
  age : integer;
  case payclass: paytype of
    salaried: (monthlyrate : real;
              startdate : integer);
    hourly: (rateperhour : real;
             reghours : integer;
             overtime : integer);
  end;
```

Type tag

```
begin employee.id:=001234;
employee.dept:=12;
employee.age:=38;
employee.payclass:=hourly;
employee.rateperhour:= 2.75;
employee.reghours := 40;
employee.overtime:=3;
writeln(employee.rateperhour,
employee.reghours,
employee.overtime);
{this should bomb as there is no
monthlyrate because
payclass=hourly}
writeln(employee.monthlyrate);
```

Output:
2.750000E+00 40 3
2.750000E+00

Pascal variant record

```

type paytype = (salaried,hourly);
var employee : record
  id : integer;
  dept: integer;
  age : integer;
  case payclass: paytype of
    salaried: (monthlyrate : real;
              startdate : integer);
    hourly: (rateperhour : real;
             reghours : integer;
             overtime : integer);
  end;

```

```

employee.payclass:=salaried;
employee.monthlyrate:=575.0;
employee.startdate:=13085;
{this should bomb as there are no
 rateperhour, etc. because
 payclass=salaried}
writeln(employee.rateperhour,
employee.reghours
employee.overtime);
writeln(employee.monthlyrate);
end.

```

Output:
5.750000E+02 13085 3
5.750000E+02

Pascal variant record - 2

```

program main (input,output);
type bits = record
  case s:integer of
    1: (i:integer)
    {say we are only using the rightmost
     byte in the word--last 8 bits}
    2: (bit: packed array [1..36] of 0..1);
  end;
var b : bits; j : integer;

```

```

begin b.s := 2;
for j:=1 to 28 do b.bit[j] := 0;
for j := 29 to 36 do b.bit[j] := 1;
writeln('initial bitstring');
for j := 29 to 36 do writeln(b.bit[j]);
writeln('b.i= ',b.i);

```

Output:

```

initial bitstring
  1
  1
  1
  1
  1
  1
  1
  1
  1
  1
b.i=      255

```

Pascal variant record -2

```

program main (input,output);
type bits = record
  case s:integer of
  1: (i:integer)
    {say we are only using the rightmost byte
     in the word--last 8 bits}
  2: (bit: packed array [1..36] of 0..1);
end;
var b : bits; j : integer;

```

```

{this should not work; change to
 type integer to implement shift
 left and shift right as arithmetic
 operations}
b.s := 1;
{simulate a left shift of one bit}
{this zeroes out all but the last 8 bits
 of the word}
b.i := (b.i * 2) mod 256 ;
writeln('left shift b.i= ',b.i);
writeln('left shift 1 bit ');
for j := 29 to 36 do writeln(b.bit[j]);

```

Output:

```

left shift b.i=      254
left shift 1 bit
 1
 1
 1
 1
 1
 1
 1
 1
 0

```

Type unions, CS314 Fall 01© BGRyder

5

Pascal variant record -2

```

program main (input,output);
type bits = record
  case s:integer of
  1: (i:integer)
    {say we are only using the rightmost byte in
     the word--last 8 bits}
  2: (bit: packed array [1..36] of 0..1);
end;
var b : bits; j : integer;

```

```

for j := 29 to 36 do writeln(b.bit[j]);
{this simulates a right shift of one
 bit}
b.i := b.i div 2 ;
writeln('right shift b.i= ');
writeln(b.i);
writeln('right shift 1 bit ');
for j:=29 to 36 do writeln(b.bit[j]);
end.

```

Output:

```

right shift b.i=
 127
right shift 1 bit
 0
 1
 1
 1
 1
 1
 1
 1
 1

```

Type unions, CS314 Fall 01© BGRyder

6