

*Read § 8.1 + 8.2 from Scott

13 Data Types - Enumeration & Subranges

Thursday, October 29, 2020 9:24 AM

Enumeration Types - Enable you to define your own type name and **literals**.

Example: Pascal

```
type weekay = (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday);
```

```
var day: weekay;
```

Pascal provides two operations on enumeration types:

```
succ(Monday) -> Tuesday
pred(Monday) -> Wednesday
```

We can write loops:

```
for today := Sunday to Saturday do
... }
```

Array Indexes:

```
var attendance : array[weekday] of integer;
...
attendance[Sunday] = 0;
```

C also has enumerated types:

```
enum weekday {Sunday, Monday, ..., Saturday};
```

```
weekday day = Sunday;
```

```
for (int i = Sunday; i <= Saturday; i++) {...}
// enums are like ints!
```

Java (enumerated types are equivalent to a class)

```
enum Weekday {Sunday, Monday, ..., Saturday};
```

```
Weekday day = Weekday.Sunday;
```

```
for (Weekday d : Weekday().values) {...}
```

Neither Python, nor Go have enumerated types.

Bigger question: Are enumerated types really useful??

Pre-defined/Built-in Types

- Numbers
 - Integers
 - Floating point
 - complex
- Characters
- Booleans
- Enumerated Types
- Subrange types

These are all **scalars**.

13 Data Types - Enumeration & Subranges

Thursday, October 29, 2020 9:24 AM

Subranges:

Pascal

```
type workday : Monday..Friday;    (* Base Type: weekday *)  
    score : 0..100;                (* Base Type: integer *)
```

Ada also has them. Mostly missing in modern programming languages.

Same Question: While they have some use, they are not considered so useful as to be included in modern programming languages.

13 Composite/Aggregates types - Records

Thursday, October 29, 2020 9:34 AM

Records/Structures

e.g. Place P_i
city, state, zip, population
 strings int

int a[10];

Pascal - record

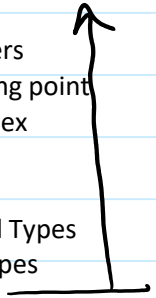
type student = record
 firstName : string;
 lastName : string;
 year : integer;
 major : string;
end;

construction mechanism

fields

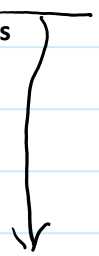
Pre-defined/Built-in Types (scalars)

- Numbers
 - Integers
 - Floating point
 - complex
- Characters
- Booleans
- Enumerated Types
- Subrange types



Composite/Aggregate Types

- ✓ Records/structures
- Arrays
- Strings
- sets
- Hash tables
- Lists
- files



① How are they defined?

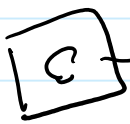
② How do I use? *positional association*

use var hp : student;
 hp := ("Harry", "Potter", 1998, "Wizardry");

firstName last year major

hp.year

access



structures / structs

```
struct student {
    char *first;
    char *last;
    int year;
    char *major;
}
```

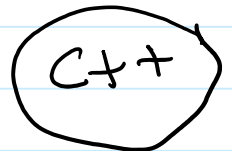
constructor

typedef <base> <newtype>;

typedef int celsius; ←

typedef struct {

```
} student;
student hp;
```



struct student hp;

hp = {"Harry", "Potter", 1998, "Wizardry"};

hp.major

use

Dip. major

13 Composite Types - Records/Structures

Thursday, October 22, 2020 9:29 AM

Go

Construct

```

type student struct {
    first string
    last  string
    year  int
    major string
}
    
```

use

named annotations

```

var hp student
hp = student{
    first: "Harry"
    last:  "Potter"
    year:  1998
    major: "Wizardry"
}
hp.year
    
```

Also

```

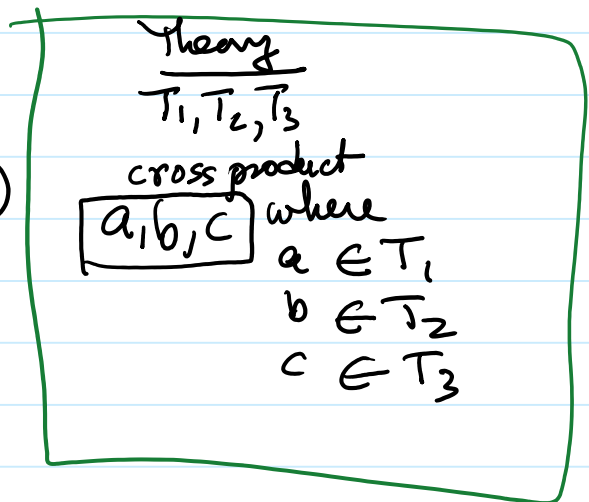
hp := student{"", "", 1998, ""}
fmt.Println(hp)
    
```

Harry Potter 1998 Wizardry

Python - Tuple

```

(a, b, c)
hp = ("Harry", "Potter", 1998, "Wizardry")
hp[0] indexed like arrays
    
```



Java

```

class student {
    String first, last, major;
    int year;
}
    
```

private

```

public student(____, _____, _____) {
    first = ____;
}

student hp = new student("#", "P", 1998, "W");
hp.year
    
```

13 Composite Types - Arrays

Thursday, October 29, 2020 9:29 AM

Arrays

Construction
mechanism

✓ int a[10]; // C
✓ int[] a = new int[10]; // Java

use

a[i] indexing

a[i:j] slicing

all elements in an array
are of the same type
[homogenous composite type]

✓ ① name - a

✓ ② type - int

→ ③ size - 10 ints

→ ④ index bounds - [0..9]

int[] a;