Structures

Objectives

At the end of the meeting, students should be able to:

- Explain the importance of structures
- Create programs using structures

Algorithms + Data Structures = Programs

- **Basic data structures**
  - Arrays, strings, arrays of strings
  - 2D arrays and beyond
  - Structures (or records)
  - Multi-level structures, arrays of structures
  - Other combinations of arrays/structures
  - Text files and other file types
- **Dynamic data structures**
  - Stacks, queues, dynamic lists
  - Trees and other hierarchical structures
  - Graphs and other network-like structures

Structures

- Structured data items can be of different types

Example:

```
typedef struct {
   char stdno[11];
   char name[30];
   int age;
} student;
```

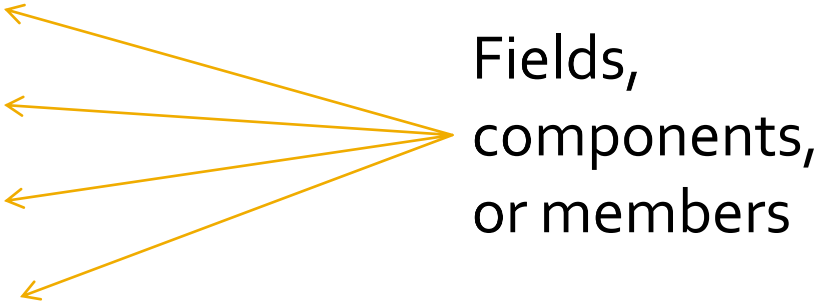
// a programmer-defined type

“2008-12345”	“CRUZ Juan”	17
s.stdno	s.name	s.age

```
student s = {“2008-12345”,“CRUZ Juan”, 17};
```

Fields can be arrays/strings …

```
typedef struct {
    char stdno[11];
    char name[30];
    int age;
    int exam[3];
} student;
```



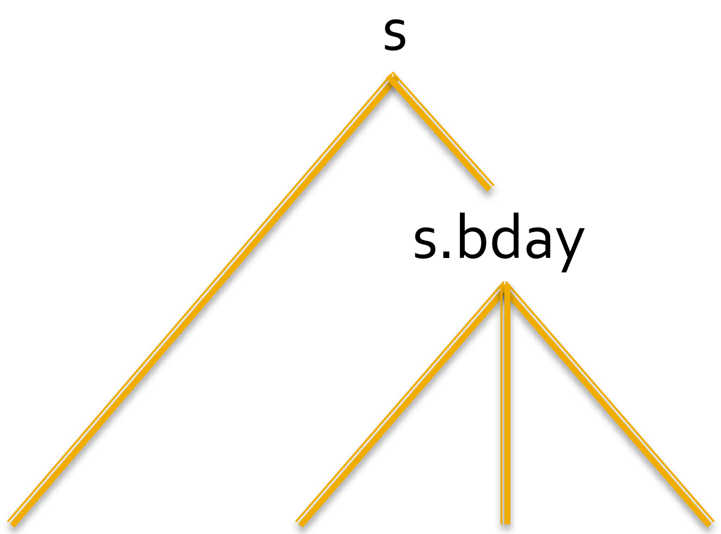
Fields, components, or members

“2008-12345”	“CRUZ Juan”	17	20	25	23
s.stdno	s.name	s.age	s.exam[0]	s.exam[1]	s.exam[2]

student s = { “2008-12345”,“CRUZ Juan”, 17, {20,25,23} };

Fields can themselves be structures ...

```
typedef struct {
   int year, month, day;
} date;
typedef struct {
   char name[10];
   date bday;
} student;
```



s

s.bday

“Juan”	1991	2	26
s.name	s.bday.year	s.bday.month	s.bday.day

```
student s = { “Juan”, {1991, 2, 26} };
```

An array of structures

```
typedef struct {
    int year, month, day;
} date;
typedef struct {
    char name[10];
    date bday;
} student;
typedef student lectureclass[100];

lectureclass yz, a, b = {
    { “Ana”, {1991, 2, 26} },     // b[0]
    {“Alice”, {1990, 4, 20} },    // b[1]
    {“Alex”, {1991, 6, 15} }      // b[2]
};
```

“Ana”	1991	2	26
“Alice”	1990	4	20
“Alex”	1991	6	15

Exercises:
Given a bigger list, write the code to print the youngest student; sort by name, sort by age; sort by birthday, etc …

Exercises for structures and arrays of structures

Give typedefs and variable declarations for

- A single hospital patient
- A list of patients
- A single subject you are enrolled in this semester
- A list of subjects you are enrolled in this semester
- A single playing card
- A pack of 52 playing cards
- Information for a single chemical element
- A list of chemical elements (periodic table)

Challenging exercise

- Given a list of subjects you have enrolled this semester, including information on class schedules, print a matrix showing your timetable

Challenging exercise

Ex:

“CMSC 11 Lect”, “9-10”, { “W”, “F” }
“CMSC 11 Lab”, “1-4”, { “Th” }
“HUM 1”, “4-5:30”, { “T”, “Th” }

	M	T	W	Th	F
9-10	---	---	CMSC 11	---	CMSC 11
1-4	---	---	---	CMSC 11	---
4-5:30	---	HUM 1	---	HUM 1	---