Computer Programming Using C

COP 3275 - Summer 2017

Lecture 22: Structures, Unions, and Enumerations (cont.)

By Ahmed E. Khaled

{/* Programming */}
Declaring a Structure Tag

• A *structure tag* is a name used to identify a *particular kind of structure*.
• The declaration of a structure tag named `part`:

```c
struct part {
    int number;
    char name[NAME_LEN+1];
    int on_hand;
};
```
Nested Structures

• Nesting one structure inside another is often useful.

```c
struct person_name {
    char first[80];
    char middle;
};
```

```c
struct student {
    struct person_name name;
    int id;
};
```

• Accessing `student1`’s first name or middle initial requires two applications of the . operator:

```c
strcpy(student1.name.first, "Fred");
student1.name.middle = 'E';
```
Arrays of Structures

• One of the most common combinations of arrays and structures is an array whose elements are structures. This kind of array can serve as a simple database.

• An array of part structures capable of storing information about 100 parts:

```c
struct part inventory[100];
```
• Accessing a part in the array is done by using subscripting:
  
  ```c
  print_part(inventory[i]);
  ```

• Accessing a member within a part structure requires a combination of subscripting and member selection:
  
  ```c
  inventory[i].number = 883;
  ```

• Accessing a single character in a part name requires subscripting, followed by selection, followed by subscripting:
  
  ```c
  inventory[i].name[0] = '\0';
  ```
struct student {
    int id;
    char name[30]
};

void main() {
    struct student record[3];
    record[0].id=1;  strcpy(record[0].name, "Raju");
    record[1].id=2;  strcpy(record[1].name, "Surendren");
    record[2].id=3;  strcpy(record[2].name, "Thiyagu");

    int l;
    for(i=0; i<3; i++){
        printf(" Records of STUDENT : %d 
", i+1);
        printf(" Id is: %d 
", record[i].id);
        printf(" Name is: %s 
", record[i].name);
    }
}
struct employee {
    char name[100];
    int age, id;
    float salary;
};

struct employee employees[10];

• Use the employee structure, develop the following methods:
• Print the employee’s id and name if the employee’s salary is between min and max value.
• Search employee array for an employee of specific id, and return the corresponding index.
Define employee structure that holds the name, ssn, salary and the date of employment. The date of employment is another structure that holds the day, month and year.

```c
struct date {
    int day;    int month;    int year;
};

struct Employee {   
    char ename[20];
    int ssn;
    float salary;
    struct date doe;
}
```