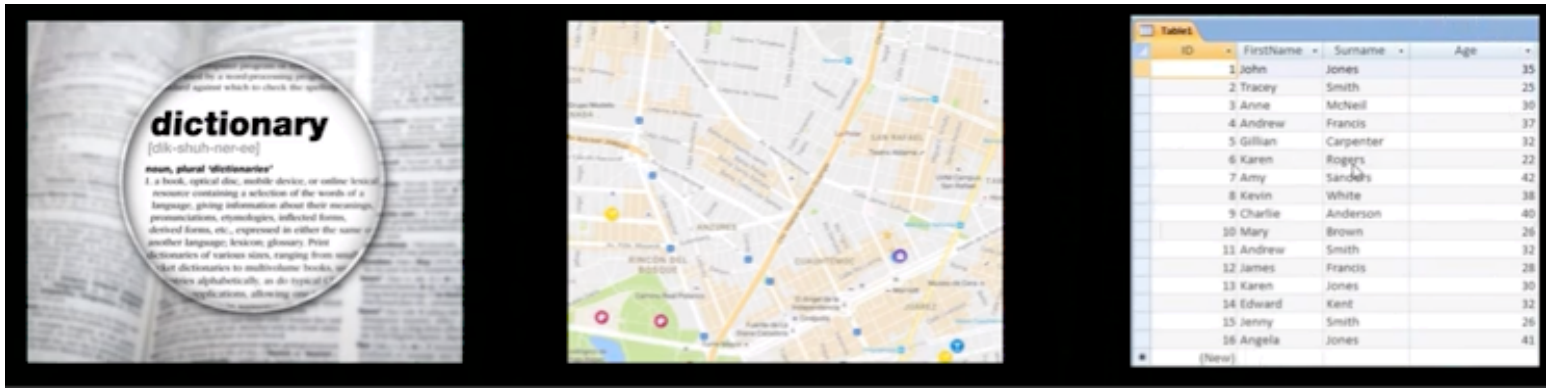
(11-1) Array of Struct
H&K Chapter 10

Instructor – Beiyu Lin
CptS 121 (June 3rd,2019)
Washington State University

WASHINGTON STATE UNIVERSITY
World Class. Face to Face.

Review of Structs



C. Hundhausen, A. O'Fallon, B. Lin

Graphs are from: https://www.youtube.com/watch?v=NzUVn8Y9deQ

Arrays of Structs (1)

- Let's first define a struct student

```
typedef struct
{
   details ...
}Type;
```

```
typedef struct
{
   int ID;
   char name[100];
   int present;
} Student;
```


C. Hundhausen, A. O'Fallon, B. Lin

Arrays of Structs (1)

- Let's first define a struct student

```
typedef struct tag
{
   details ...
}Type;
```

```
typedef struct student
{
   int ID;
   char name[100];
   int present;
} Student;
```


C. Hundhausen, A. O'Fallon, B. Lin

Arrays of Structs (1)

- Let's first define a struct student

```
typedef struct
{
    int ID;
    char name[100];
    int present;
    struct Student* next;
} Student;
```

```
typedef struct student
{
    int ID;
    char name[100];
    int present;
    struct student* next;
} Student;
```


C. Hundhausen, A. O'Fallon

Arrays of Structs (1)

- Let's first define a struct student

```
typedef struct student
{
    int ID;
    char name[100];
    int present; // Attended class or not
} Student;
```

- Next we will build up an attendance sheet

C. Hundhausen, A. O'Fallon

Arrays of Structs (2)

```
int main (void)
{
    Student attendance_sheet[100]; /*100 students in the class*/

    return 0;
}
```

declare structure variable for Student

- Let's look at a logical view of this attendance sheet on the next slide

C. Hundhausen, A. O'Fallon

Arrays of Structs (3)

- Attendance sheet, which consists of multiple struct student types

0	1	2	…	99
{ID, name, present}	{ID, name, present}	{ID, name, present}	…	{ID, name, present}
1000	1108	1216		10692

C. Hundhausen, A. O'Fallon

Arrays of Structs (4)

- To initialize one item in the array, try:

```
attendance_sheet[index].ID = 1111;
strcpy (attendance_sheet[index].name, “Bill Gates”);
Attendance_sheet[index].present = 1;
   // 1 means in attendance, 0 means not in present
```


C. Hundhausen, A. O’Fallon

Conclusions

- Each item in a list of structs represents one struct
- You may manipulate an array of structs using indices and the dot member operator

C. Hundhausen, A. O'Fallon

References

- J.R. Hanly & E.B. Koffman, *Problem Solving and Program Design in C (8th Ed.)*, Addison-Wesley, 2016
- P.J. Deitel & H.M. Deitel, *C How to Program (7th Ed.)*, Pearson Education , Inc., 2013.

Collaborators

- Chris Hundhausen
- Andrew O'Fallon