CSC209 Week 6 Notes

Hyungmo Gu

May 14, 2020

Struct 1 of 3

- Introducing Structs
  - **struct/structures** is like dictionary in Python or object in Javascript
  - there are differences between array and structure

	array	structure
data of same type	yes	not required
declaration details	type and number of elements (array [] notation)	types of members (struct keyword)
access via ...	index notation	dot notation

  - items in struct is called **member**
  - items in array is called **element**

```c
#include <stdio.h>
#include <string.h>

int main() {
    struct student {
        char first_name[20];
        char last_name[20];
        int year;
        float gpa;
    };

    struct student good_student;
```

```
13
14          strcpy(good_student.first_name, "Jo");
15          strcpy(good_student.last_name, "Smith");
16          good_student.year = 2;
17          good_student.gpa = 3.2;
18
19          printf("Name: %s %s\n", good_student.first_name, good_student.
    last_name);
20          printf("Year %d. GPA %.2f\n", good_student.year, good_student.
    gpa);
21
22          return 0;
23      }
24
```

Listing 1: struct_example_1.c

Struct 2 of 3

- Using Structs in Functions
  - Array pass function by **reference** (of the pointer of first element).
    - Changing value inside affects outside
  - Struct pass function by **value** like int and string.
    - Changing value in function doesn't affect value outside
    - Pointer used to pass by **reference**

```
1   #include <stdio.h>
2   #include <string.h>
3
4   struct student {
5       ...
6   };
7
8   void change(struct student *s) { // <- passes by
    reference
9       ...
10  };
11
12  int main(void) {
13      struct student good_student;
14      ...
15      change(&good_student); // <- to pass function by
    reference (This is too cool!!!)
16      ...
17      return 0;
18  }
19
```

Listing 2: struct_example_2.c