Week 3

Structs : user-defined aggregate of different data types that is accessed by a single name.

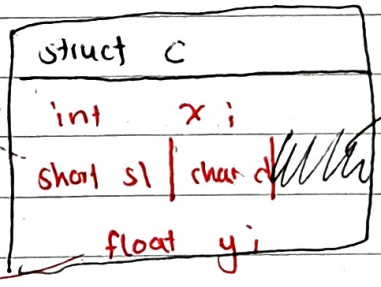
```
struct Person {
    char name[50];
    int age;
};

struct Person p1 = {"Isaac", 21};

typedef struct Person person_t;
```

Now, we can create instance like below

```
person_t p2 = {"John", 23};
```

Memory Alignment (depends on ~~alignment~~ architecture)

```
struct c {
    int x;
    short s1;
    char c1;
    float y;
};
```



padding
{ can be removed by compiler flags

Union: structs for more efficient memory usage

union's size: largest attr's size.

```
struct catalog {
    char title[];
    enum catalogType type
    union Creator { } info
};

union creator =
{ .info = _____ }

struct book {
    char * author
    char * ISBN
} book_info

struct film {
    char * director
    char * producer
} film_info
```

BITwise operators

- shifts <<, >>
- &, |, ^, ~
  - & — and
  - | — or
  - ^ — xor
  - ~ — not

Files

file name

file mode (r, w)

```
FILE *fp = fopen("hi.txt", "r")
```

read

~~while(!feof(stdin)){ int nread = fgets(...)~~

buf length

```
while( fgets(buf, 64, stdin) != NULL) {

}
```

write

```
fprintf( stderr , "%s", "invalid");
```

- stderr — or stdout
- format