

Simple data structure to store different data types in one container

ex.1 :  

```
typedef struct Student {
    char *name;
    int id;
} Stud;
```

```
int main() {
    Stud student1; char student1.name = malloc(10);
    student1.name = "Ahmed";
    printf("%s", student1.name);
}
```

This previous example shows the way to define and access a struct named Stud

ex.2 :  

```
typedef struct Student {
    char *name;
    int id[20];
} Stud;
```

char \*name : It is just a reference for array of chars  
 But the advantage that you can change its size dynamically

int id[20] : Array of integers but has fixed size

```
ex.3: typedef struct Student {  
    char *name;  
    int id;  
} S;
```

```
void modify(S *s1);
```

```
int main(void) {  
    S s1;  
    modify(&s1);  
}
```

```
void modify(S *s1) {  
    s1->name = "Ahmed"; // accessing members of struct  
    s1->id = 1;  
}
```

This example shows that it is preferred to pass a struct by its reference

so you can directly modify its members by lambda expression

(->) ↵



## Padding

// normal struct definition

```
typedef struct s {
    int a;
    char b;
    int c;
} abc;
```

size of (abc) = 12

a	a	a	a	b	E	E	E	c	c	c	c
First CPU cycle				Second CPU cycle				Third CPU cycle			

What matter: reduction of CPU cycles  
when accessing one member

What doesn't matter: memory saving

## Packing:

#pragma pack(1)

```
typedef struct s {
    int a;
    char b;
    int c;
} abc;
```

sizeof (abc) = 9

a	a	a	a	b	c	c	c	c
First CPU cycle				second CPU cycle				3rd CPU cycle

What matters: Memory saving

What Doesn't matter: no. of CPU cycles

Structs: Container of same or different kind of data types

- not able to include functions

Objects: It is a blueprint of class

- has all the characteristics of OOP
- Not found in C (just C++)
- Can have method (functions)