



دانشگاه بین المللی امام خمینی قزوین  
دانشکده فنی و مهندسی

# مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

## فصل دوازدهم: ساختارها در C

نستوه طاهری جوان

[nastoooh@aut.ac.ir](mailto:nastoooh@aut.ac.ir)



## ساختمان

○ یک `structure` یک ساختمان داده برای نگهداری عناصر با نوع های مختلف است.

• نحوه تعریف ساختمان:

```
struct  نام ساختمان {
    نوع  نام فیلد 1;
    نوع  نام فیلد 2;
    ....
    نوع  نام فیلد n ;
    نام متغیرها
}
```

• نوشتن نام ساختمان و نام متغیرها اختیاری است.



## ساختمان

○ مثال: یک ساختمان برای اطلاعات دانشجویان

```
struct student{  
    char name [20];  
    char family [20];  
    float mark;  
    long int id;  
} std1, std2, std3;
```

این ساختمان سه فیلد برای نام، فامیل و نمره دارد.

سه متغیر از نوع این ساختمان تعریف شده است که هر متغیر، هر چهار فیلد تعریف شده را دارند.



## ساختمان

○ برای دسترسی به فیلدهای هر متغیر از عملگر **نقطه** استفاده می کنیم.

```
struct student{  
    char name [20];  
    char family [20];  
    float mark;  
} std1, std2, std3;
```

...

```
std2.mark = 12.5;  
strcpy (std3.name, "ALI");  
scanf ("%f", &std1.mark);  
gets (std3.family);
```



## ساختمان

○ میتوان متغیرهای از نوع ساختمان را بعد از تعریف ساختمان تعریف کرد.

```
struct student{  
    char name [20];  
    char family [20];  
    float mark;  
};  
  
...  
  
struct student std1, std2;
```



## ساختمان

- حتی میتوان برای یک ساختمان، یک نوع داده جدید تعریف کرد.
- برای تعریف نوع داده جدید از typedef استفاده می کنیم.

```
struct student{  
    char name [20];  
    char family [20];  
    float mark;  
};  
  
typedef struct student Student;  
  
...  
  
Student std1, std2;
```



## ساختمان

○ آرایه ای از ساختمان (توضیح با مثال)

```
struct student{  
    char name [20];  
    char family [20];  
    float mark;  
};
```

```
struct student std [20];
```

```
std [1]. mark = 13.6;  
scanf ("%f", &std[5].mark);  
std[3].name[5] = 'A';  
strcpy (std[0].family, "Taheri");
```

حرف ششم از اسم دانشجوی چهارم  
را تغییر میدهد.



## ساختمان

○ ساختمان تو در تو (توضیح با مثال)

```
struct date{
    int year;
    int month;
    int day;
};
struct student {
    char name [20] ;
    char family [20];
    struct date b_d;
};

struct student st[20];

st[2].b_d.year = 1375;
st[0].b_d.day = 25;
strcpy (st[5].name, "ALI");
```

به نحوه ارجاع به فیلدها دقت کنید!!





## ساختمان

○ اشاره گر به ساختمان (توضیح با مثال)

```
struct student {  
    char name [20] ;  
    char family [20];  
    long int id;  
};  
struct student st, *p;  
  
p = &st;  
  
(*p).id = 123456;  
p->id = 123456;
```

به نحوه ارجاع به فیلدها دقت کنید!!  
برای این کار دو روش وجود دارد.



## ساختمان

○ ساختمان خود ارجاع

```
struct student {  
    char name [20] ;  
    char family [20];  
    long int id;  
    struct student *pst;  
};
```

فیلد چهارم این ساختمان، یک اشاره گر به یک عنصر از همین جنس است.

○ این نوع ساختمان در آینده کاربردهای زیادی در برنامه نویسی برای شما پیدا خواهد کرد.

• برای درک مثال های این ساختمان، مفهوم "لیست پیوندی" را جستجو کنید.



## ساختمان

○ تمرین مهم:

- درباره مفاهیم زیر تحقیق کنید:
  - مفهوم کلاس.
  - مفهوم برنامه نویسی شی گرا.



پایان