جلسه ی دوازدهم (استراکچر ها قسمت اول)

اگر مبحث آرایه هارو به یاد داشته باشد برای ذخیره لیست لیست اسامی از یک آرایه ی دو بعدی کاراکتری استفاده می کردیم حالا فرض کنید هر اسمی یک نمره هم داره و می خوایم این نمره هار وهم کنار اسامی ذخیره کنیم اولین ایده اینه که یک ارایه از نوع floatداشته باشیم که عناصر یکم تا انم آن با لیست اسامی مرتبط می شه خب تااینجای قضیه هیچ مشکلی نیست ام حالا فرض کنید بخوایم این لیست رو بر اساس نام مرتب کنیم این مطلب واضح است که باید برای مرتب سازی عناصر آرایه رو جابجاکنیم اما مشکل اینه که باید برای هر نفر هم اسم و هم نمره فرد رو جابجاکنیم تا نمره کسی با نمره دیگری عوض نشه خب می گیم باشه قبول چند تا دستور باید بیشتر بنویسیم دیگه!! غیر از اینه... اما اگه فیلد هامون بیشتر از دوتا بود چی ؟؟ هممون می دونیم که بانک های اطلاعاتی معمولا بیشتر از دو فیلد داده ای دارن . برای نوشتن دستور های جابجایی رکودها برای مثلا یک رکورد ده فیلدی باید ۳۰ خط کد بنویسیم خب حالا یه مشکل دیگه برای فرستادن یه رکورد به یه تابع برای مثلا ده فیلد باید تابعمون ۱۰ آرگمان داشته باشه (بعد فردا پس فردا تو روزنامه ها مینویسن فردی بر اثر کد نویسی بش از اندازه جان باخت!!!!) و اینطوری شد که چیزی به نام استراکچر (structure)ساخته شد.

تعریف:

استراکچر امکاناتی است که به ما اجازه ساخت انواع داده ای جدید را از ترکیب انواع داده ای مو جود می دهد.

با استفاده از این امکانت ما می توانیم چندین متغیر از انواع دیگر را در یک بسته بندی بعنوان نوع جدیدی معرفی کنیم به هر یک از این متغیر های درون این بسته بندی فیلد و به هر بسته یک اسراکچر یا به علت شباهت زیاد به یک رکورد, رکورد می گویند. بعد از تعریف یک استراکچربعنوان یک نوع داده ای جدید می توانیم بااین بسته ی متغیر ها مثل یک متغیر معمولی رفتار کنیم و هر زمان که نیاز بود محتویات هر یک از متغییر های درونش یا همان فیلد ها را تغییر دهیم

نحوه تعریف یک استراکچر:

```
struct  نام نوع داده ای جدید{
تعاریف فیلدها
};
```

مثال:

```
struct person{
 char name[٤٠];
 float grade;
};
```

در این مثال استراکچری بانام personتعریف شده پس از این به بعد نوع داده ای جدیدی به این نام داریم که می توانیم متغیر هایی از آن تعریف کنیم برای مثال

```
person p١;
```

یا آرایه ای از آن تعریف کنیم

```
Person list[٢٠];
```

مقدار دهی اولیه به استراکچر ها شبیه مقدار دهی اولیه به آرایه هاست

```
person p١={"Ali Ahmadi",٩,٧٥};
person list[٢٠]={{"Ali Ahmadi",٩,٧٥},{"Reza Amini",١٠,٠}};
```

خب حالا ما متغیر هایی داریم که نام و نمره افراد رو تحت یک عنوان و یک متغیر در یک بسته بندی ذخیره می کنند اگر بخواهیم محتویات این متغیر هارا جابجا کنیم در عوض

cpplearning.blogfa.com

جابجایی تک تک فیلد ها فقط استراکچر هارا جابجا می کنیم برای مثال می خواهیم آرایه ی لیستی را که چند خط قبل تعریف کردیم عناصر صفرم و یکمش را جابجا کنیم

```
person tmp;
tmp=list[۰];
list[۰]=list[۱];
list[۱]=tmp;
```

به همین را حتی اول یک متغیر کمکی گرفتیم و بعد عناصر نام برده را جابجا کردیم!!!
برای آرگمان توابع هم دیگه نیازی نیست به تعداد فیلدها آرگمان تعریف کنیم فقط یک آرگمان از نوع استراکچر مورد نظر تعریف میکنیم
تا اینجا توانستیم چند متغیر را از انواع مختلف را در یک بسته بندی قراردهیم و بدون اینکه نامی از محتویات بسته ها ببریم , بسته هارا جابجا کنیم و در یکدیگر کپی کنیم ام بالاخره همه این ها برای این است که بتوانیم محتویات بسته هارا بهتر استفاده کنیم پس مسلمه که نیاز به دسترسی به محتویات این بسته ها یا همان فیلده داریم برای این موضوع از عملگر نقطه استفاده میکنیم

مثال:

```
p۱.name
P۱.grade
list[۰].name;
list[index].grade;
```

عبارات فوق برای دسترسی به نام و نمره فرد در استراکچریست که قبلاً تعریف کردیم عبارات فوق دقیقاً از نوع داده ای فیلدهای نام برده هستند یعنی آرایه ی کاراکتر و ممیز شناور و دقیقاً مثل انواع نام برده با آنها رفتا می شود برای مثال دستور های زیر کاملاً مجاز هستند مثل انوع داده ای معمولی!!!

```
cin>>p۱.name; خواندن آرایه ی کاراکتر
cin>>p۱.grade; خواندن ممیز شناور
cout<<p۱.name;(آرایه ی کاراکتر) چاپ رشته
cout<<p۱.grade;چاپ ممیز شناور
cin>>list[۰].name; خواندن آرایه ی کاراکتر برای نام نفر اول لیست
cin>>list[index].grade;خواندن ممیز شناور برای نمره ی نفر اندیسم لیست
```

جلسه ی آینده به بررسی استراکچر ها بعنوان آرگمان توابع و اشاره گر های آنها می پردازیم بعد سعی میکنم را جع search ,sortصحبت کنم تا چیز های که تاحالا خواندیم بخصوص استرا کچر ها یا همان ساختمان ها رو کار بردی یاد بگیریم