جلسه نهم(چپ مقدارها و راست مقدار ها (L_Value,R_Value

مطلب خیلی ساده است در هر عبارت انتساب عبارتی که می تواند سمت چپ قرار گیرد یک چپ مقدار است و عبارتی که می تو اند سمت را ست قرار گیرد راست مقدار است و ای توانای بدین معناست که صحت عبارت را از بین نبرد در این مثال xچپ مقدار وyراست مقدار است

```
x = y;
```

در حالت کلی تمام متغیر ها و به عبارتی از دید شئ گرایی تمام اشیا چپ مقدارند و تمام ثوابت رشته ای ,کاراکتری و عددی چه به صورت شناسه و چه به صورت تایپ مستقیم راست مقدارند. توجه داشته باشید که یک عبارت می تواند هم چپ مقدار و هم راست مقدار باشد

برای مثال ثوابتی که در برنامه تعریف میشوند در لحظه تعریف که مقدار اولیه می گیرند چپ مقدارند اما بعد از آن راست مقدارند چرا که دیگر نمی توانند مقدار جدیدی بگیرند

تمام متغیر ها هم می توانند چپ مقدار باشند هم راشت مقدار

هدف از بیان این اصطلاحات خسته کننده دونکته مهم است
اول
رفرنس ها برای گرفتن مقدار اولیه نیاز به چپ مقدار دارند
به مثال توجه کنید

```
نوع داده ای &نام متغیر =چپ مقدار;

int& x= a++;
int& x= ۷۸;
int& x=a/۲;
```

هیچ یک از سه مثال فوق صحیص نمی باشند زیرا هیچ یک از عبارات سمت راست چپ مقدار نیستند
اما مثال های زیر صحیح می باشند

```
int& x=a;
int& x=++a;
```

توابع دارای نوع بازگشتی راست مقدارند و نمی توانند در سمت چپ عبارات قرار گیرند مثل ;٤=()myfunc اما راه حلی برای این کار وجود دارد حقیقت این است که برخی مواقع استفاده این چنینی از یک تابع می تواند خیلی مفید باشد برای انجام این کار باید نوع باز گشتی تابع را به عنوان رفرنس تعریف نمود
مثال معروف این مبحث

```
int& max(int& a,int& b){
  if(a>b)
   return a;
  else
   return b;
}

void main(){
 int x=۳,y=٤;
 max(x,y)=۱۰۰;
```

cplearning.blogfa.com

```
cout<<max(x,y);
}
```

دراین مثال بدون دانستن اینکه xبزرگتر است یا yمتغییر بزرگتر را تغییر دادیم و چاپ نمودیم!

چگونه کار میکند؟

برای اجرا چنین توابعی خود را گیج نکنید مثل توابع معمولی ابتدا تابع را اجرا نمایید بعد بجای عبارت فرا خوانی تابع یک رفرنس معمولی در نظر بگیرید که این رفرنس دقیقاً رفرنس عبارت باز گشت داده شده توسط دستور returnدرون تابع است به همین سادگی
پس مسیر عکس این را در پیش نگیرید مثل این که عدد ۱۰۰را بخواهید در تابع کپی کنید!!!

نکته:

اگر به مثال بالا دقیق تر نگاه کنید می بیند که آرگمان های تابع هم از نوع رفرنس هستن این نکته هیچ ربطی به رفرنس بودن نوع باز گشتی نداره یعنی اینکه می توان توابعی ایجاد کر که نوع بازگشتی رفرنس داشته باشن و آرکمان ها شون رفرنس نباشن پس این دو هیچ ربطی بهم ندارن اما دلیل این کار اینه که اگر آرگمان ها آرگمان معمولی باشند متغییر محلی محسوب میشوند و بعد از پایان اجرا تابع از بین میروند حالا اگر قرار باشد تابع رفرنسی به یکی از این متغییرها برگرداند مقدار باز گشتی تابع رفرنسی خواهد بود به محل نامعلومی از حافظه از آن فراتر تابع کار نخواهد کرد اما باتعریف اینگونه آرگمان ها پارامتر ها خود رفرنسی به متغیر های برنامه اصلی است و از حا فظه آنها (x,y)استفاده میکنند بنابر این بعد از پایان اجرا تابع حا فظه مربوط به آنها آزاد نخواهد شد. پس همیشه باید رفرنس باز گشتی تابع رفرنسی به متغیری باشد که بعد از پایان اجرا تابع حافظه اش آزاد نشود(به سیستم برگردانده نشود).

یک مثال معروف دیگر دسترسی به خانه های آرایه با استفاده از تابع است این مثال وقتی با مباحث شی گرایی مطرح می شود به کونه می شود که با دو عبارت x[index]وfunc(index) هم معنی می شوند و کاملا یکسان عمل میکنند این مثال را در مباحث شی گرایی مطرح می کنیم.

جلسه ی آینده در مورد رشته ها بحث می کنیم

cpplearning.blogfa.com