آموزش C++ - درس ۸: مقدمه ای بر اشاره گرها

اشاره گرها در C و C++ متغیرهایی هستند که آدرسها را در خود نگه می دارند. آنها قدرت بیشتری را برای برنامه نویس فراهم می کنند و به او در دسترسی به داده ها و تغییر آنها از روشهایی که در بسیاری از زبانهای برنامه نویسی دیگر دیده نمی شود، کمک می کنند. همچنین آنها برای ارسال پارامترها به توابع مفید هستند به صورتی که اجازه می دهد یک تابع تغییرات لازم را انجام داده و مقادیر را به روال فراخواننده برگشت دهد. وقتی که از آنها بصورت نادرست استفاده شود، منبع خطاهای برنامه و اشکالات برنامه نویس خواهند بود.

مقدمه:
وقتی یک برنامه اجزا شده باشد، همه متغیرها در حافظه ذخیره شده اند، هر کدام از آنها یک آدرس یا محل منحصربفرد دارند. عموما یک متغیر و آدرس حافظه مربوط به آن حاوی مقادیر داده است. برای مثال، وقتی شما یک متغیر بصورت زیر تعریف می کنید:

```
int count = 5;
```

مقدار ۵ در حافظه ذخیره شده است و با استفاده از متغیر count قابل دسترسی است. یک اشاره گر نوع خاصی از متغیر است که به جای مقدار داده حاوی آدرس حافظه است. همانطور که در زمان استفاده از متغیر معمولی آن را تغییر می هید، مقدار آدرس ذخیره شده در یک اشاره گر با دستکاری متغیر اشاره گر تغییر می کند. معمولا آدرس ذخیره شده در یک اشاره گر، آدرس متغیرهای دیگر است.

```
تعریف یک اشاره گر به int:
int *ptr;
ذخیره آدرس متغیر count در اشاره گر ptr:
ptr = &count;
عملگر یگانی & آدرس یک متغیر را بر می گرداند
```

برای بدست آوردن مقدار موجود در محلی از حافظه که در اشاره گر ذخیره شده است، نیاز به عمل "بازگرداندن مرجع" داریم. بازگرداندن مرجع با استفاده از عملگر یگانی * انجام می شود.

```
int total;
مقدار موجود در آدرس ذخیره شده در ptr به متغیر total اختصاص داده می شود
total = *ptr;
```

بهترین راه برای یادگیری چگونگی استفاده از اشاره گرها استفاده از مثال است. در اینجا مثالهایی از انواع اعمالی که در مورد آنها صحبت شد آورده شده است. اشاره گرها یک موضوع مشکل به حساب می آیند. اگر هنوز مطلب برایتان روشن نشده است، عجله نکنید.

اعلان و مقداردهی اولیه
اعلان و مقداردهی اولیه اشاره گرها نسبتا ساده است.

```
int main()
{
   int j;
   int k;
   int l;
                              یک اشاره گر به int تعریف می کند:
   int *pt1;
                              یک اشاره گر به int تعریف می کند:
   int *pt2;
   float values[100];
   float results[100];
                           یک اشاره گر به float تعریف می کند:
   float *pt3;
                           یک اشاره گر به float تعریف می کند:
   float *pt4;
   j = 1;
   k = 2;
                  آدرس متغیر j را در اشاره گر pt1 قرار می دهد:
   pt1 = &j;
                  آدرس متغیر k را در اشاره گر pt2 قرار می دهد:
   pt2 = &k;
آدرس اولین عنصر از مقادیر را د ر اشاره گر pt3 قرار می دهد:
   pt3 = values;
```

```
                                    این عبارت معادل عبارت بالایی است:
    pt3 = &values[0];

    return 0;
}
```

باز گرداندن مرجع اشاره گر/ تخصیص مقدار

باز گرداندن مرجع اجازه تغییر در داده موجود در آدرسی از حافظه که در اشاره گر ذخیره شده است را می دهد. اشاره گر یک آدرس حافظه را ذخیره می کند. باز گرداندن مرجع اجازه می دهد که آدرس موجود در آن آدرس حافظه تغییر داده شود. عملگر یگانی * برای بازگرداندن مرجع بکار می رود.

```
*pt1 =*pt1 + 2;
```

این عبارت دو تا به مقدار اشاره شده در pt1 اضافه می کند. بله این عبارت دو تا به محتویات آدرس حافظه موجود در اشاره گر pt1 اضافه می کند. بنابراین با توجه به برنامه اصلی، pt1 حاوی آدرس متغیر j است. متغیر j به ۱ مقداردهی شده بود. تاثیر عبارت فوق اضافه کردن ۲ تا به متغیر j است.

محتوای آدرس موجود در اشاره گر ممکن است به یک اشاره گر دیگر و یا یک متغیر تخصیص داده شود.

```
تخصیص محتوای حافظه اشاره شده بوسیله اشاره گر pt1 به محتوای حافظه اشاره شده بوسیله اشاره گر Pt2:
*pt2 = *pt1;
                                    تخصیص محتوای آدرس اشاره شده بوسیله اشاره گر pt2 به متغیر k:
k = *pt2;
```

محاسبات اشاره گر

قسمتی از قدرت اشاره گرها مربوط به توانایی انجام محاسبات در خود اشاره ها است. اشاره گرها می توانند با استفاده از عبارات محاسباتی اضافه شوند، تفریق شوند و یا دستکاری شوند. اشاره گر pt3 و آرایه از نوع float به نام values را به خاطر بیاورید.

```
                    آدرس اولین عنصر آرایه values در اشاره گر pt3 ذخیره می شود:
pt3 = &values[0];
                    با اجرای عبارت زیر pt3 به دومین عنصر آرایه values اشاره می کند:
pt3++;
                    مقدار عدد Pi در عنصر دوم آرایه values تخصیص می یابد:
*pt3 = 3.1415927;
                    با اجرای عبارت زیر اشاره گر pt3 به عنصر ۲۷ ارایه values اشاره می کند:
pt3 += 25;
                    مقدار عنصر ۲۷ آرایه برابر مقدار 2.22222 قرار می گیرد:
*pt3 = 2.22222;
                    با اجرای عبارت زیر pt3 به ابتدای آرایه values اشاره می کند:
pt3 = values;
                    با اجرای حلقه for زیر مقدار همه عناصر آرایه به 37.0 تخصیص می یابد:
for (ii = 0; ii < 100; ii++)
{
    *pt3++ = 37.0;
}
                    عبارت زیر آدرس اولین عنصر آرایه values را در pt3 قرار می دهد:
pt3 = &values[0];
                    عبارت زیر آدرس اولین عنصر آرایه results را در pt4 قرار می دهد:
pt4 = &results[0];

for (ii=0; ii < 100; ii++;)
{
      عبارت زیر محتوای محل حافظه ای که آدرس آن در pt3 ذخیره شده است را در
محتوای محل حافظه ای  که آدرس آن در pt4 ذخیره شده است، تخصیص می دهد.
  *pt4 = *pt3;
    pt4++;
    pt3++;
}
```

ترجمه: محمد صافدل
منبع: http://cplus.about.com