C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

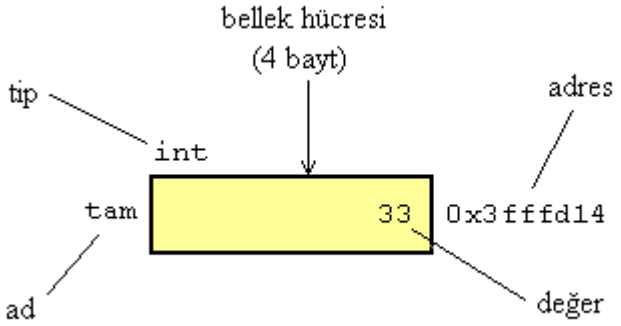
İşaretçiler (Pointers)

İşaretçiler yani pointerlar C++ dilinin en temel yapıtaşlarından biridir. İşaretçiler konusu için C++'ı diğer dillerden ayıran en önemli özelliklerinden biridir. Adından da anlayabileceğiniz üzere C++'ta işaretçiler değişkenlerin bellekteki adreslerini tutabilme ve onlarla işlem yapabilme olanağı sağlar

Belli bir tipte değişken tanımlanıp ve bir değer atandığında, o değişkene dört temel özellik eşlik eder:

- değişkenin adı
- değişkenin tipi
- değişkenin sahip olduğu değer (içerik)
- değişkenin bellekteki adresi

Örnek olarak bir tamsayı değişken tanımlayalım
int tam = 33;
Int tanımlandığındanbellekte 4 bayt büyüklüğünde bir yer ayrılır.



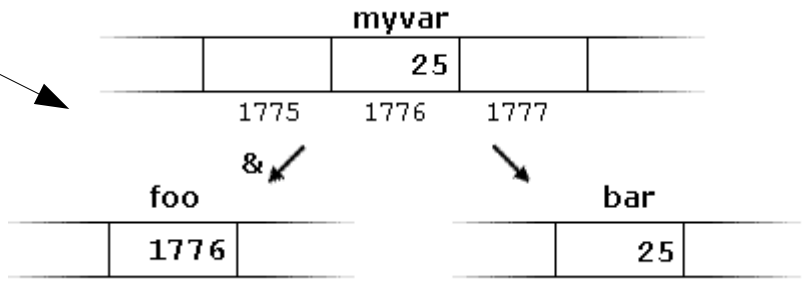
4 baytlık alanda değişkenin binary değeri:
00000000 00000000 00000000 00100001

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

Adres & operatörü

foo = &myvar;
Bu işlem ile myvar değişkeninin adresi foo değişkenine atanır.
Örnek olarak myar değişkeninin hafızada 1776 nolu adreste olduğunu düşünelim. Aşağıdaki kod parçasında yapılan işlem

```
myvar = 25;
foo = &myvar;
bar = myvar;
```



C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

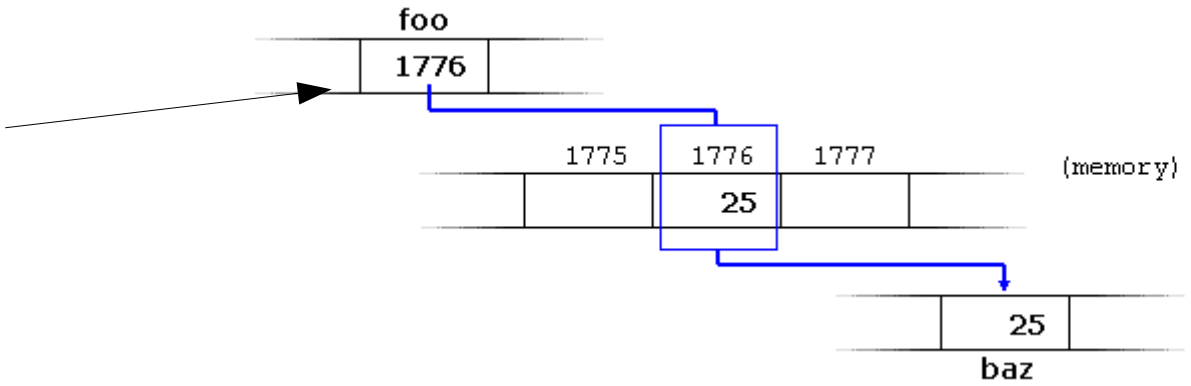
"*"POİNTER (GÖSTERİCİ, İŞARETÇİ)

Bir veri bloğunun bellekte bulunduğu adresi içeren (gösteren) veri tipidir.
Tanımlama biçimi:

veri tipi *p;

p değişkeni <veri tipi> ile belirtilen tipte bir verinin bellekte saklandığı adresi içerir.

Örnek:
baz = *foo;



baz = foo; // baz ve foo eşdeğer (bellek adresleri aynı 1776)
baz = *foo; // baz foo'nun işaret ettiği değere eşit (25)

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

& ——►operetörün işaret ettiği hafızadaki adresi belirtir

* ——► operatörü işaret ettiği hafızadaki değeri

İşaretçi (gösterici), hafıza alanındaki bir adresinin saklandığı değişkendir. İşaretçilere veriler (yani değişkenlerin içeriği) değil de, o verilerin bellekte saklı olduğu yerin başlangıç adresleri atanır. İşaretçi adres tutan bir değişkendir.

Bir işaretçi, diğer değişkenler gibi, sayısal bir değişkendir. Bu sebeple kullanılmadan önce program içinde bildirilmelidir. İşaretçi tipindeki değişkenler şöyle tanımlanır:

```
Tip  *işaretçi_adı;
```

Burada Tip herhangi bir C tip adı olabilir. Değişkenin önünedeki * karakteri yönlendirme (indirection) operatörü olarak adlandırılır ve bu değişkenin veri değil bir adres bilgisi tutacağını işaret eder. Örneğin:

```
char  *ch;            /* tek bir karakter için */

int   *x;             /* bir tamsayı için */

float *deger, sonuc;   /* deger işaretçi tipinde, sonuc sıradan bir gerçel değişkenler
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek1:

```
#include <iostream>
using namespace std;
int main ()
{
  int ilksayi, ikincisayi;
  int * mypointer;
  mypointer = &ilksayi;
  *mypointer = 10;
  mypointer = &ikincisayi;
  *mypointer = 20;
  cout << "İlk Sayi =  " << ilksayi << '\n';
  cout << "İkinci Sayi = " << ikincisayi << '\n';
  return 0;
}
```

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 2:

```
#include <iostream>
using namespace std;
int main ()
{
  int ilksayi = 5, ikincisayi = 15;
  int * p1, * p2;
  p1 = &ilksayi;  // p1 = ilksayi değ. adresi
  p2 = &ikincisayi; // p2 =  ikincisayi değ. adresi
  *p1 = 10;         // value pointed to by p1 = 10
  *p2 = *p1;        // value pointed to by p2 = value pointed to by p1
  p1 = p2;          // p1 = p2 (value of pointer is copied)
  *p1 = 20;         // value pointed to by p1 = 20

  cout << "İlk sayi :: " << ilksayi << '\n';
  cout << "İkinci Sayi :: " << ikincisayi << '\n';
  return 0;
}
```

Çıktısı:

```
İlk sayi :: 10
İkinci Sayi :: 20
```

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

İşaretçiler ve Diziler:

Diziler aşağida gösterildiği gibi uygun tipteki işaretçilere dönüştürülebilir

```
int myarray [20];
int * mypointer;
mypointer = myarray;
```

Bir dizinin, i. elemanına erişmek için *(mypointer+i) işlemi yapılması zorunludur. Yani

```
*mypointer+i;   /* mypointer nin gösterdiği değere (dizinin ilk elemanına) i sayısını ekle */
*(mypointer+i); /* mypointer nin gösterdiği adresten i blok ötedeki sayıyı hesapla
```

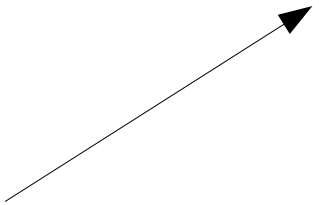
*/anlamındadır. Çünkü, * operatörü + operatörüne göre işlem önceliğine sahiptir

Örnek :

```
// more pointers
#include <iostream>
using namespace std;
int main ()
{
  int numbers[5];
  int * p;
  p = numbers;  *p = 10;
  p++;  *p = 20;
  p = &numbers[2];  *p = 30;
  p = numbers + 3;  *p = 40;
  p = numbers;  *(p+4) = 50;
  for (int n=0; n<5; n++)
    cout << numbers[n] << ", ";
  return 0;
}
```

Çıktısı

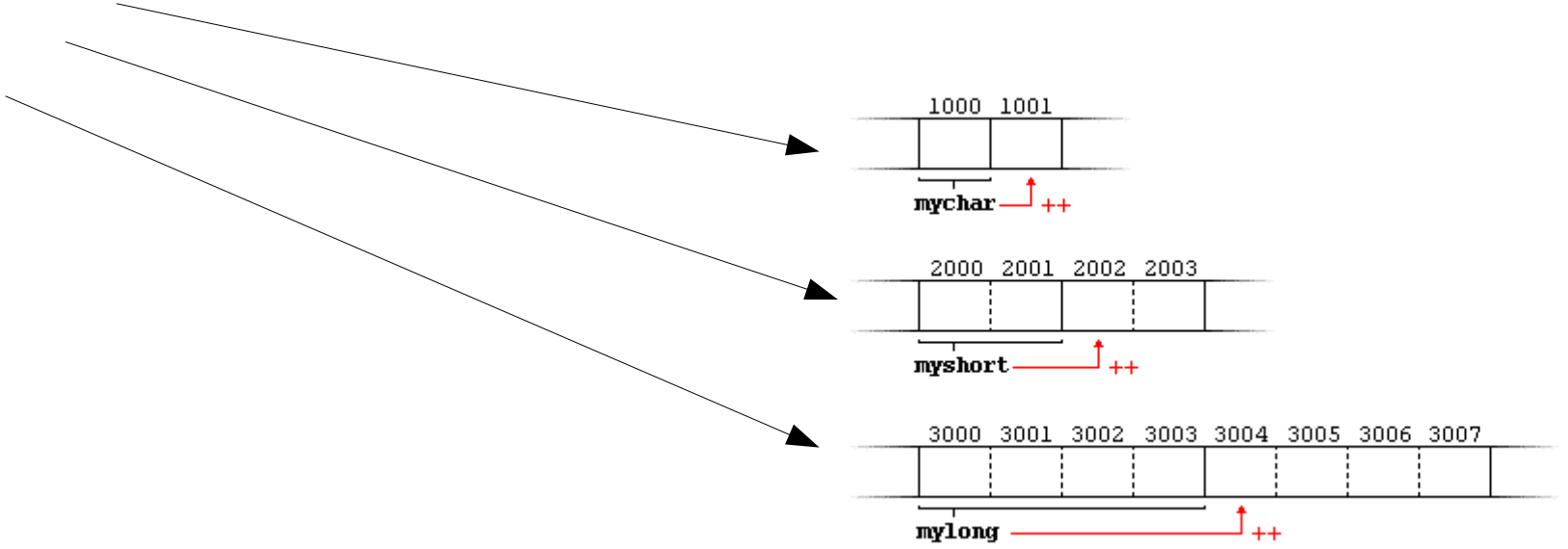
```
10, 20, 30, 40, 50,
```



C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

İşaretçiler ile hesap yapma:

Örnek olarak aşağidaki gibi üç farklı tipte işaretçimiz olsun:
char *mychar;
short *myshort;
long *mylong;
(Hafızada char 1 byte short iki bayte long 4 byte büyüklüğünde yer kaplar)
Bu işaretçilerin hafızadaki yerlerinin sırasıyla 1000, 2000, ve 3000 olsun ve bu işaretçiler ile şu işlemleri yapalım
++mychar;
++myshort;
++mylong;



Aşağıdaki kullanımda eşdeğer sonuç verir
mychar = mychar + 1;
myshort = myshort + 1;
mylong = mylong + 1;

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek:

```
*p++   // == *(p++):  işaretçide artış, and arttırılmamış  addresi gösterme
*++p   // == *(++p):  işaretçide artış, and arttırılmış  addresi gösterme
++*p   // ==++(*p): işaretçi, ve onun işaret ettiği değerde  artış
(*p)++ // işaretçi, ve post-increment the value it points to
```

```
// pointers as arguments:
#include <iostream>
using namespace std;
void increment_all (int* start, int* stop)
{
  int * current = start;
  while (current != stop) {
    ++(*current);  // increment value pointed
    ++current;     // increment pointer
  }
}
void print_all (const int* start, const int* stop)
{
  const int * current = start;
  while (current != stop) {
    cout << *current << '\n';
    ++current;     // increment pointer
  }
}
int main ()
{
  int numbers[] = {10,20,30};
  increment_all (numbers,numbers+3);
  print_all (numbers,numbers+3);
  return 0;
}
```

MİMAR SİNAN
FINE ARTS
UNIVERSITY

C++ Giriş Ders 8
MSGSU Fizik Bölümü
Ferhat ÖZOK