

## 7.2. Dinamik bellek kullanımı

Daha önceki derslerimizde yerel ve global değişkenlerden bahsedilmişti. Yerel değişkenler tanımlandıkları blok içinde veya fonksiyon içinde geçerli olup blok veya fonksiyon sonlandığında bellekten otomatik olarak atılan, yok edilen değişkenlerdir. Global değişkenler ise ana program sonlanıncaya değin bellekte yer işgal etmeyi sürdürürler. Bu değişkenlerle her hangi bir işlem yapılmasa dahi bellekten atılması mümkün değildir.

Bazen bellekten istediğimiz zaman istediğimiz kadar(ihtiyacımız kadar) yer alma ve işimiz bittiğinde de aldığımız yerleri serbest bırakmak isteyebiliriz. Tabi ki burada yer alma ve serbest bırakma işlemi program çalışırken(run time) gerçekleşmektedir. Belleğin bu şekilde kullanılması dinamik bellek kullanımı olarak tanımlanır.

C dilinde bellekte yer alma için `malloc( )`, ayrılan yeri geri vermek için `free( )` fonksiyonları kullanılmaktadır. C++ da ise dinamik bellek işlemlerini gerçekleştirmek için daha çok yeteneğe sahip `new` ve `delete` komutları kullanılır. `new` bellek ten yer almak için, `delete` ise alınan yerleri geri vermek için kullanılır. Aşağıda verilen örneği inceleyiniz.

```
int *p;
p=new int;      //Tamasayı bir değer için bellekten yer al
*p=100;         //Ayrılan yerin(adresin) içine 100 değeri at
delete p;       //ayrılan yeri serbest bırak.(geri ver)
```

Örnekler:

```
float *p;
p=new float;
*p=3.14;
delete p;
```

`int *p=new int;` ifadesi aşağıdaki iki satırın yerine geçer.

```
int *p;
p=new int;
```

`int *p=new int(100);` ifadesi aşağıdaki üç satırın yerine geçer. Adres alındıktan sonra içine başlangıç değeri olarak 100 değeri atılmaktadır.

```
int *p;
p=new int
*p=100;
```

Dinami bellek kullanımı ile ilgili olarak:

`int *p=new int[10];` ifadesi 10 elemanlı `int` türünden bir dizi için bellekte yer ayırmakta ve başlangıç adresini `p` ye atamaktadır.

Dizi elemanlarına sanki normal bir diziyi kullanıyormuşçasına `p[0]=5; p[1]=2;` şeklinde erişmek mümkün olmaktadır.

Daha sonra bu dizi için ayrılan yeri

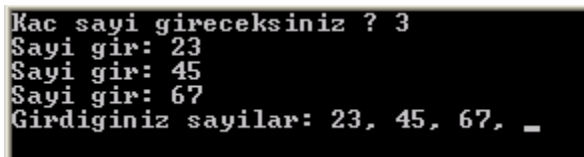
**delete [ ] p;**

ifadesi ile geri vermek mümkün olmaktadır. Bu ifadede dizi sembolü [ ] kullanımına dikkat ediniz.

Verilen örnek programda klavyeden istenen değer kadar yer ayrılmakta ve girilen değerler ekrana yazdırılmaktadır.

```
,
#include <iostream.h>
#include <conio.h>
using namespace std;
int main ()
{
    int i,n;
    int * p;
    cout << "Kac sayi gireceksiniz ? ";
    cin >> i;
    p= new int[i];

    if (p == 0)
        cout << "Hata: bellekten yer alinamadi";
    else
    {
        for (n=0; n<i; n++)
        {
            cout << "Sayi gir: ";
            cin >> p[n];
        }
        cout << "Girdiginiz sayilar: ";
        for (n=0; n<i; n++) cout << p[n] <<" , "; //for döngüsü içinde sadece cout var.
        delete[] p;    //bu komut for dongusune dahil değil
    }
    getch();
    return 0;
}
```



```
Kac sayi gireceksiniz ? 3
Sayi gir: 23
Sayi gir: 45
Sayi gir: 67
Girdiginiz sayilar: 23, 45, 67, _
```