

Cihan Hoca 2

class Person

```
{
    Person& GoSchool ()
    {
        //Referans döndürmeye çalışıyor. Objenin
        kendisini döndürmeye çalışıyor. Fonksiyon bittikten sonra obje silinecek.
        //firstMeet stack'de yaratıldı.
        //firstMeet heap'de yaratılsaydı, bu seferde
        heap'dekini silme işlemi sıkıntılı.
        Person firstMeet;
        ...
        return firstMeet;
    }
}
```

int main ()

```
{
    Person me;
    ...
    Person fm = me.GoSchoool ();
}
```

****Stack'de** yaratılan her şey silinir, function call için.

****MoveConstructor** bir bak.

****sharedPointer** (bir objeye erişen pointerların sayısını tutuyor, pointer silme işlemini kafandan at.)

& referans, kopyalama işlemi olmaması için.

vector class'ı, dizi(array) ile ilgili işlemler yapılıyor. (exponential olarak büyüyor.)

Template ile kullanılıyor.

push.back()	sona eleman eklemek için.
pop.back()	sondan bir şey eklemek için.
vec [1]	aradan bir elemanı getirmek ve üstüne yazabilmek için.
erase (1, 3)	birinci elemandan başlayarak 3 tane sil.
remove (3)	üçüncü elemanı

```
vector < hangi template ile çalışması gerektiğini söylüyorsun >  
vector < double >  
vector < string >
```