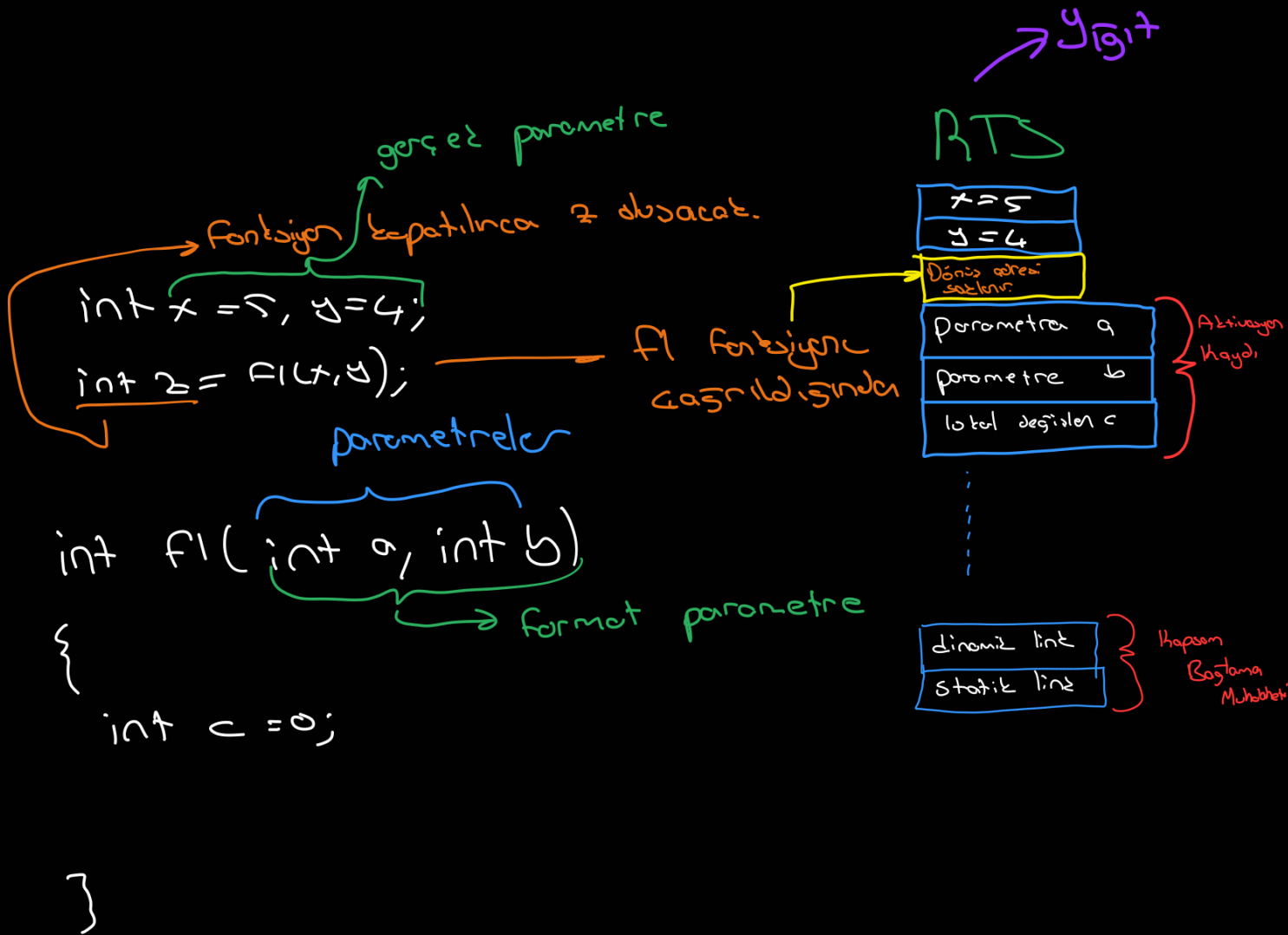


# AI+ Programlama / Modüler Programlama

## Modül: Fonksiyon, Sınıf



# Parametre Aktarma Sistemleri?

## Değer ile Gösterme

```
void degistir(int a){  
    a=100;  
}
```

gösteren

```
int x=5;  
degistir(x);  
print(x); (5)
```

gösteren

RTD

x	5
---	---

a	100
---	-----

→ Aktarılan

keydi

→ fonksiyon

kenarına siliniyor

## Adres ile Gösterim

```
void degiştir (int *p) {
```

\*p = 100;

p'nin gidiş adresi

}

RTS

0x00E

x	100
r	0x00E

Aktarılan Keyfi

p	0x00E
---	-------

Silinecek

```
int x=5;
```

```
int *r = &x;
```

```
degiştir(r);
```

```
print x: (100)
```

Referans ile Çağırma  $\rightarrow C++, C\#$

Formal Parametre için yer ayrılmaz.

void degistir(int &a) {  $\rightarrow$  içerisinde int den bir değişkeni referans alıyor.

a = 100;

a bir değişken. Sim. a altında x'i gösteriyor. a içindeki x'i her değişiklik direkt x'i etkiliyor.

}

int x = 5;

degistir(x);

print(x);

---

void degistir(int \*p) {  $\rightarrow$  Bu da referans ile

p = new int(500);  $\rightarrow$  çağırma. içerisinde adres olan bir değişkeni referans alıyor. Belleğin

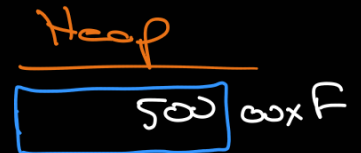
}

int a = 5;

int \*r = a;

degistir(r);

print a;



## Sonuç ile Çalışma C #

```
void topla (int a, int b, out int c) {
```

Sonuç parametresi  
C a or b den  
olduktan sonra  
return'den önce değeri  
almalı

```
    c = a + b;  
}
```

```
int x=10, y=20, z;
```

```
topla(x, y, out z)
```

Fonksiyon dışı C'nin aldığı  
değeri buraya gelecektir.

```
void topla (int a, int b, int &C) {
```

C++ benzeri;

```
    C = a + b;  
}
```

```
int x=10, y=20, C;
```

```
topla(x, y, C)
```

C burada referans olarak gider  
ama aynı değişkeni göstermektedir.

