

## Veri Tipleri

- 1- İlkel Veri Tipleri – Sayfa 50,51,52
- 2- Tür dönüşümü, işlemler
- 3- Pointer Gösterge Tipi
  - a. Adres ve değer ilişkisi

```
#include <iostream>
using namespace std;

int main(){
    int a=10;
    cout<<a<<endl;
    cout<<&a<<endl;
    return 0;
}
```

### b. Pointer Tanımlama

```
int main(){
    int a=10;
    int *p;
    p=&a;
    cout<<a<<endl;
    cout<<&a<<endl;
    cout<<p<<endl;
    cout<<*p<<endl;
    cout<<&p<<endl;
    return 0;
}
```

### c. Değer atama

```
int main(){
    int a=10,b;
    b=a;
    a++;
    cout<<a<<endl;
    cout<<b<<endl;
    return 0;
}
```

### d. Değer atama

```
int main(){
    int a=10;
    int *p;
    p=&a;
    a++;
    cout<<*p<<endl;
    return 0;
}
```

```
int main(){
    int a=10;
    int *p,*r;
    p=&a;
    r=p;
    *r=50;
    cout<<a<<endl;
    return 0;
}
```

```

int main(){
    int a=10;
    int *p,*r;
    p=&a;
    r=p;
    a=100;
    cout<<*r<<endl;
    return 0;
}

```

e. Eşitlik Kontrol

```

int main(){
    int a=10,b=10;
    if(a == b) cout<<"Esit";
    else cout<<"Esit degil";
    return 0;
}

```

```

int main(){
    int a=10,b=10;
    int *p,*r;
    p=&a;
    r=&b;
    if(p == r) cout<<"Esit";
    else cout<<"Esit degil";
    return 0;
}

```

#### 4- Double Pointer

```

int main(){
    int a=50;
    int *p=&a;
    int **pp = &p;
    cout<<"pp nin adres: "<<&pp<<endl;
    cout<<"pp nin degeri: "<<pp<<endl;
    cout<<"pp nin gosterdigi: "<<*pp<<endl;
    cout<<"p nin adres: "<<&p<<endl;
    cout<<"p nin degeri: "<<p<<endl;
    cout<<"p nin gosterdigi: "<<*p<<endl;
    cout<<"a nin adres: "<<&a<<endl;
    cout<<"a nin degeri: "<<a<<endl;
    return 0;
}

```

#### 5- Sınıf Pointer

```

class Sayi{
private:
    int deger;
public:
    Sayi(int dgr){
        deger = dgr;
    }
    Sayi(){
        deger=0;
    }
    friend ostream& operator<<(ostream& ekran,Sayi& sag){
        ekran<<sag.deger<<endl;
        return ekran;
    }
};

```

```

int main(){
    Sayi *s1,*s2,s3;
    // Kac adet nesne var?

    return 0;
}

```

## 6- Sınıf Pointer devam

```
int main(){
    Sayi *s1,*s2,s3;
    s1 = &s3;
    s2 = &s3;
    cout<<*s1<<endl;
    cout<<*s2<<endl;
    cout<<s3<<endl;
    return 0;
}
```

## 7- Heap Bellek Bölgesi

```
int main(){
    Sayi *s1 = new Sayi(100);
    delete s1;
    return 0;
}
```

```
int main(){
    Sayi *s1 = new Sayi(100);
    Sayi *s2 = s1;
    delete s1;
    return 0;
}
```

## 8- Sallanan Gösterici

```
int main(){
    Sayi *s1 = new Sayi(100);
    Sayi *s2 = s1;
    delete s1;
    Sayi *s3 = new Sayi(500);
    cout<<*s2;
    return 0;
}
```

## 9- Yıkıcı Fonksiyon

```
class Sayi{
private:
    int *pdeger;
public:
    Sayi(int dgr){
        pdeger = new int(dgr);
    }
    Sayi(){
        pdeger = new int(0);
    }
    friend ostream& operator<<(ostream& ekran,Sayi& sag){
        ekran<<*sag.pdeger<<endl;
        return ekran;
    }
    ~Sayi(){
        delete pdeger;
    }
};
```

```
int main(){
    Sayi *s1 = new Sayi(100);
    cout<<*s1;
    delete s1;
    return 0;
}
```

## 10- Swap işlemi

```
int main(){
    Sayi *s1 = new Sayi(100);
    Sayi *s2 = new Sayi(200);
    Sayi *syedek;
    syedek=s1;
    s1=s2;
    s2=syedek;
    cout<<*s1;
    cout<<*s2;
    return 0;
}
```

11- NULL pointer, Ekrana sıfır yazacaktır.

```
int main(){
    Sayi *s1 = NULL;
    cout<<s1;
    return 0;
}
```

## 12- Fonksiyon değer adres ve referans ile çağırma özet

<pre>void Degistir(int a){     a=100; } int main(){     int x=200;     Degistir(x);     cout&lt;&lt;x;     return 0; }</pre>	<pre>void Degistir(int &amp;a){     a=100; } int main(){     int x=200;     Degistir(x);     cout&lt;&lt;x;     return 0; }</pre>	<pre>void Degistir(int *p){     *p=100; } int main(){     int x=200;     int *r=&amp;x;     Degistir(r);     cout&lt;&lt;x;     return 0; }</pre>	<pre>void Degistir(int *&amp;p){     p=NULL; } int main(){     int x=200;     int *r=&amp;x;     Degistir(r);     cout&lt;&lt;*r;     return 0; }</pre>
--	---	---	---

## 13- Örnek – Bellekte oluşum haritasını göster

```
class Sayi{
private:
    int *pdeger;
public:
    Sayi(int dgr){
        pdeger = new int(dgr);
    }
    Sayi(){
        pdeger = new int(0);
    }
    friend ostream& operator<<(ostream& ekran,Sayi& sag){
        ekran<<*sag.pdeger<<endl;
        return ekran;
    }
    ~Sayi(){
        delete pdeger;
    }
};
class Kisi{
private:
    string isim;
    Sayi *yas;
```

```

public:
    Kisi(string isim,int ys){
        this->isim = isim;
        yas = new Sayi(ys);
    }
    ~Kisi(){
        delete yas;
    }
};
int main(){
    Kisi *ahmet = new Kisi("Ahmet",35);
    Kisi *mehmet = new Kisi("Mehmet",55);

    delete ahmet;
    delete mehmet;
    return 0;
}

```

14- Yıkıcı fonksiyona çağrıldı yazıp delete sil ve çağrılmadığını göster.

15- Kisi yas public kısma al ve yaşları değiştir.

```

...
friend ostream& operator<<(ostream& ekran,Kisi& sag){
    ekran<<sag.isim<<" "<<*sag.yas<<endl;
    return ekran;
}
...

int main(){
    Kisi *ahmet = new Kisi("Ahmet",35);
    Kisi *mehmet = new Kisi("Mehmet",55);
    cout<<*ahmet;
    cout<<*mehmet;
    Sayi *yasYedek = ahmet->yas;
    ahmet->yas = mehmet->yas;
    mehmet->yas = yasYedek;
    cout<<*ahmet;
    cout<<*mehmet;

    delete ahmet;
    delete mehmet;
    return 0;
}

```