

* Stream state:

Her stream'in sağıt durumunu belirten bir state'i vardır. Eger stream hatalı durumdaya bürünse onun gibi olası nedenlerden yarınca olur.

```
int main()
{
    using namespace std;

    ios::iostate
```

→ bu adaların mostler ile geleneklidir.

* Type Değişkenler:

```
int main()
{
    using namespace std;

    cout << boolalpha;
```



```
cout << "cin.good() = " << cin.good() << "\n";      ↗ Ayni
cout << "cin = " << (cin ? true : false) << "\n";      ↗ end of file = false
cout << "cin.eof() = " << cin.eof() << "\n";           ↗
cout << "cin.fail() = " << cin.fail() << "\n";          ↗ Ayni
cout << "!cin = " << !cin << "\n";                   ↗ Ayni
cout << "cin.bad() = " << cin.bad() << "\n";
```



Hatalı bir stream olursa:

```
Microsoft Visual Studio Debug Console
bir sayı girin: ali → Zayıf string verince olusan
                                hata
cin.good() = false
cin = false
cin.eof() = false
cin.fail() = true
!cin = true
cin.bad() = false → bad, format hatası, dosyayı
                                okumakta başarısız. Sıvı format hatası
D:\CONCURRENCY\PACA_2022\Release\PACA_2022.e
Press any key to close this window . . .
```

→ Akımda hatalı bir dosya → eof
→ Stream tam yüklü beklenen → bad
yoksa vs...

* Clear:

```
int main()
{
    using namespace std;
    int x;

    std::cout << "bir tam sayı girin: ";
    cin >> x;

    print_stream_state(cin);
    cin.clear(); → Tam olumsuz durumları reset
                    eder, stream kullanılabilir
                    tekrardan → good state
```

```

int main()
{
    using namespace std;

    cin.clear(ios::failbit | ios::eofbit); ]→ Alıştır degerler set eder
    I

    cin.setstate(); ]→ Alıştır degerler "vega" lar, eules
    print_stream_state(cin);

}

```

* ignore:

- Eger argumentisiz çağrılırsa → streamdeki 1 karakteri ignorer / streamden silerdir
- " argumenti → argument → 0 sağ. kodlu karakter ignorer / streamden silerdir
bir sağı

```

int main()
{
    using namespace std;

    int x;

    cout << "bir sayı girin: ";
    cin >> x;
    cout << "bir sayı daha girin: ";
    cin.ignore(5, '\n');

    ↳ 5 elemen ignore'la ye da 5 karakter gelmeden  
newline gelince ignore ignarula

}

```

```

int main()
{
    using namespace std;

    istringstream iss{ "ali56798veli34512" };

    int x;

    iss.ignore(_Count:3);
    iss >> x;

    std::cout << "x = " << x << "\n";
    iss.ignore(_Count:4);
    int y;

    iss >> y;
    std::cout << "y = " << y << "\n";
}

```

Microsoft Visual Studio Debug Console
 x = 56798
 y = 34512
 D:\CONCURRENCY\PACA_2022\Release\PACA_2022.exe (pr
 Press any key to close this window . . .

* Dosya İşlemleri: #include<fstream>

→ Bu basitlik dosyasiyle birlikte 3 basit dosyasi eklenis durur

std::ifstream	→ dosyadan okuma	→ fstream
std::ofstream	→ dosyaye yazma	→ fstream
std::fstream	→ her iki de	→ fstream

* Büt dosya nesnesinin nesne olmasi, o nesnenin holdenin bir dosya ile ilişkilendirildigi anlamına gelir!

```
int main()
{
    using namespace std;

    ofstream ofs; → default initialized, first, herhangi bir class'a bagis edilebilir
    cout << (ofs ? "good" : "failed") << "\n";
    ofs state good
}
```

```
int main()
{
    using namespace std;
    ofstream ofs;
    if (ofs.is_open()) {
        cout << "acik dosya var\n";
    }
    else {
        cout << "acik dosya yok\n";
    }
}
```

```
int main()
{
    using namespace std;
    ofstream ofs{ "volkan.txt" }; → dosya acilmis

    if (ofs.is_open()) {
        cout << "acik dosya var\n"; ✓
    }
    else {
        cout << "acik dosya yok\n"; ✓
    }

    ofs.close(); → dosya kapatildi, ornek kod dosya yok!
    if (ofs.is_open()) {
        cout << "acik dosya var\n";
    }
    else {
        cout << "acik dosya yok\n"; ✓
    }
}
```

* ifstream ve ofstream, RAII iddian: temel gorul bir örnek

```
{ → scope sonuna gelince, ofs'nin bogisi teromlenir.
    ofstream ofs{ "volkan.txt" };
    //file operations
}
```

→ scope sonuna gelince, ofs'nin bogisi teromlenir.
Ofs'nin destructor'i cagrilidiginda → Dosya da kapanır!

* ios::openmode: → Veri turu dosyaya bagli, bitmask

ios::in	→ okuma
ios::out	→ yazma
ios::app	→ son eklemme (append)
ios::trunc	→ dosya silinir
ios::ate	→ dosya acilir adam file pointer sona
ios::binary	→ binary mode

→ Dosya türleri nesnesinin constructor'ina 2. argument olarak geçirilir.iger gecilirse, default olarak ios::out | ios::trunc

0 rumlu dosya
vergi o dosya girmesi.

```

int main()
{
    ifstream ifs{ "out.txt", ios::ate };

```

* Append modinde: file pointer sonda degil!!
Yazma istenilen okuma sende!

```

if (ifs.fail()) { } → stream olsadı hata var mı?
    cerr << "dosya acilamadi\n";
    return 1;
}

cout << "dosya acildi\n";
cout << "file pointer : " << ifs.tellg() << "\n";

```

File pointer'in
konumunu
gösterir.

ios::ate oldugu için
en sondan

* ifstream:

```
using namespace std;
```

int main()

```
//fopen("ali.txt", "r") ] C'deki fopen ("m, 'r')
ifstream ifs{"ali.txt", ios::in} 
```

korrigile bular!

int main()

```
//fopen("ali.txt", "r+")
fstream fs{"ali.tx"} 
```

"r+" otomatik olarak, ama istenirse
dosyaye yonlur. ios::in | ios::out

int main()

```
fstream fs{ "suleyman.txt", ios::in | ios::out | ios::trunc}; 
```

C'dındaki fopen w +

* open fonksiyon ile, dosya üzerinde yonitimsız ifstream nesnesinin okuma/gizme attributeleri degistirilebilir. Sonradan degistirelebilir.

```
using namespace std;
```

int main()

```
ifstream ifs;
    I yonetildi
    /**
    ifs.open("ali.txt", ios::ate);
    //
    ifs.close();
        dosya kapatildi
    ifs.open("veli.txt", ios::binary)
```

Daha sonra dosya acildi

open mode degistiriliip caldi.

using namespace std;

```
1 std::ofstream create_text_file(const std::string& filename)
2 {
3     std::ofstream ofs{filename};
4     if (!ofs) {
5         throw std::runtime_error{filename + " cannot be created!"};
6     }
7     return ofs;    X value expression'a dosyası! Tasima semantiği devreye girer.
8 }
```



```
1 std::ifstream open_text_file(const std::string& filename)
2 {
3     std::ifstream ifs{filename};
4     if (!ifs) {
5         throw std::runtime_error{filename + " cannot be opened!"};
6     }
7     return ifs;
8 }
```

*Tipik Dosya İşlemleri:

dosyadan okuma işlemleri
 formatted input → left shift cc
 unformatted input → get
dosyaya yazma işlemleri ← *yapın ab iletti / Randıslı*
dosya konum göstericisi (file pointer)

* Örnekler:

```
// ilk 1 milyon asal sayiyi primes.txt isimli dosyaya
// bir satırda 10 asal sayı olacak şekilde
// yazınız
```

```
int main()
{
    std::ofstream ofs{ "primes.txt" };

    if (!ofs) {
        std::cerr << "dosya olusturulamadi\n";
        return 1;
    }

    int prime_count{};
    constexpr int no_of_primes = 10'000;
    int x{ 2 };
    while (prime_count < no_of_primes) {
        if (isprime(x)) {
            if (prime_count && prime_count % 10 == 0)
                ofs << '\n';
            ofs << x << " ";
            ++prime_count;
        }
        ++x;
    }
}
```

* Text mode vs Binary mode:

- Text mode'de $20 \rightarrow \text{eof}$
- Text mode'da newline $\rightarrow 2$ karakter kopyol. (13 ve 10 karakteri)
- Binary mode'dan newline $\rightarrow 1$ karakter kopyol (sadece 10)

* Dosyalar Formatı Okuma Örneği:

```
1  #include <fstream>
2  #include <iostream>
3  #include <ostream>
4  #include "nutility.h"
5  #include "date.h"
6  #include <iomanip>
7  #include <random>
8  #include <format>
9  #include <conio.h>
10
11
12
13
14 int main()
15 {
16     using namespace std;
17
18     auto ifs{ open_text_file(filename: "records.txt") };
19
20     int id;
21     string name, surname, town;
22
23     string town_entry;
24
25     cout << "sehri girin: ";
26     cin >> town_entry;
27
28     while (ifs >> id >> name >> surname >> town) {
29         if (town == town_entry)
30             cout << std::format(_Fmt:"{} {} {} {}\n", &_Args:id, &_Args:name, &_Args:surname, &_Args:town);
31     }
32
33
34
35
36 }
```

I

verdict function template

id name surname town

faklı formatlananın 43 tane yerineki argümanları.

* Osistem Reader:

```
1 int main()
2 {
3     set<Date> myset;
4
5     for (int i = 0; i < 10000; ++i) {
6         myset.insert(Date::random());
7     }
8
9     cout << myset.size() << "\n";
10
11    auto ofs{ create_text_file("tarihler.txt") };
12
13    int mon_day;
14    std::cout << "ayin gununu girin: ";
15    cin >> mon_day;
16
17    copy_if(myset.begin(), myset.end(), ostream_iterator<Date>{ofs, "\n"}, [mon_day](const Date& date) {
18        return date.month_day() == mon_day;
19    });
20
21
22
23
24
25
26
27
28
29
30
31
32
33 }
```