

# Algoritmalar ve Programlama II – BLM 102

## Hafta 11: İstisna İdaresi (Exception Handling)



Fenerbahçe Üniversitesi

# Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: [emre.levent@fbu.edu.tr](mailto:emre.levent@fbu.edu.tr)

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 311

Email: [ugur.ozbalkan@fbu.edu.tr](mailto:ugur.ozbalkan@fbu.edu.tr)

# Ders Planı

- İstisna İdaresi
- Try-Catch-Throw Mekanizmaları

# İstisna İdaresi (Exception Handling)

- Bir istisna (exception), kodun çalışma zamanında hata durumunda oluşan bir olaydır.

# İstisna İdaresi (Exception Handling)

- İstisna olayları yaşandığında, istisna idaresi mekanizmaları ile program istisna yaşandığında çalıştırılması istenen kod parçacıklarına atlar.
- C++ dilinde istisnaları yakalamak için, 3 keyword kullanılmaktadır.
  - Try
  - Catch
  - Throw

# İstisna İdaresi (Exception Handling)

- İstisna olayları yaşandığında, istisna idaresi mekanizmaları ile program istisna yaşandığında çalıştırılması istenen kod parçacıklarına atlar.
- C++ dilinde istisnaları yakalamak için, 3 keyword kullanılmaktadır.
  - Try
  - Catch
  - Throw

# İstisna İdaresi (Exception Handling)

- Try – Catch-Throw:

Bu bloğun içerisinde istisnai durum yaşandığında, kodun akışı istisna idarecisine devredilmektedir.

Bir istisnai durum yaşanmadığında kod normal akışından devam eder.

```
try {  
    // Hata oluşturabilecek kod parçacıklarını  
    throw exception; Hata oluşturma  
}  
catch () {  
    // Hatayı yakalayan catch bloğu  
}
```

# İstisna İdaresi (Exception Handling)

- Try – Catch-Throw:

Try bloğun içerisinde oluşturulan istisnalar, catch bloğu tarafından yakalanmaktadır.

Try bloğunun içerisinde istisnalar throw keyword'ü ile oluşturulmaktadır.

```
try {  
    // Hata oluşturabilecek kod parçacıklarını  
    throw exception; Hata oluşturmumu  
}  
catch () {  
    // Hatayı yakalayan catch bloğu  
}
```



# İstisna İdaresi (Exception Handling)

```
#include<iostream>
using namespace std;
int main() {

try {
int yas = 15;
if (yas > 18) {
cout << "Yetki Verildi";
}
else {
throw (yas);
}
cout << "Kod istisna üretmedi\n";
}
catch (int myNum) {
cout << "Yetki verilmedi, 18 yas alti\n";
cout << "Yas: " << myNum;
}
return 0;
}
```

Çıktı:

Yetki verilmedi, 18 yas alti

Yas: 15

# İstisna İdaresi (Exception Handling)

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

double bolum(int a, int b) {
    if (b == 0) {
        throw "0'a bolum tespit edildi!";
    }
    return (a / b);
}
```

```
int main() {

    try {
        int sayi1 = 10;
        int sayi2 = 0;
        double sonuc = bolum(sayi1, sayi2);
        cout << "Bolum sonucu: " << sonuc;
    }
    catch (const char* e) {
        cout << "Istisna olustu: " << e;
    }

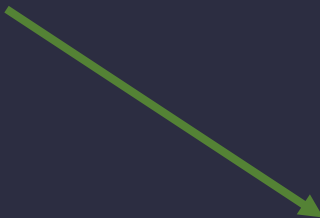
    return 0;
}
```

Çıktı:

Istisna olustu: 0'a bolum tespit edildi!

# İstisna İdaresi (Exception Handling)

- Try-catch bloğunda, throw ile gönderilen exception'un türüne göre catch'a argüman türü yazılmalıdır.



```
catch (const char* e) {  
    cout << "İstisna olustu: " << e;  
}
```

# İstisna İdaresi (Exception Handling)

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

int main() {
try
{
throw 20;
}
catch (int e)
{
cout << "İstisna olustu: " << e << "\n";
}
return 0;
}
```

İstisna olustu: 20

# İstisna İdaresi (Exception Handling)

```
try {  
    // kodlar  
}  
catch (int param) {  
    cout << "int exception";  
}  
catch (char param) {  
    cout << "char exception";  
}  
catch (...) {  
    cout << "Diğer exceptionlar";  
}
```

# İstisna İdaresi (Exception Handling)

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

int main() {
try
{
int* myarray = new int[1000];
}
catch (exception& e)
{
cout << "Standard exception: " << e.what() << endl;
}
return 0;
}
```

C++ Exception  
kütüphanesi,

Çalışma zamanında  
bellekte yeterli yer  
olmadığında istisna  
oluşturma

İstisna	Açıklama
<code>bad_alloc</code>	Veri tahsisi yapılırken hata
<code>bad_cast</code>	Veri türü dönüşümü hatası