

## الأول: يعني إيه Exception؟

ال Exception ببساطة هو غلط بيحصل وقت تشغيل البرنامج (runtime)، زي مثلاً:

تحاول تقسم على صفر.

تفتح فايل مش موجود.

تدخل داتا تايب مش متوقعة.

تخرج برا boundaries بتاعت اراي.

لما الغلط ده يحصل، البرنامج العادي بيوقع (crash) ويقف. لكن إحنا مش عايزين ده يحصل في برنامجنا.

هنا بييجي دور ال Exception Handling

ال Exception Handling هو إنك تحط كودك في safe zone، بحيث لو حصل أي غلط، البرنامج:

يمسك الغلط.

يتصرف بشكل كويس (مثلاً يطبع رسالة، أو يتجاهل الغلط، أو يجرب حل ثاني).

في C# عندك 3 كلمات مفتاحية أساسية:

بتحط فيها الكود اللي ممكن يبوذ: try:

بيتمسك هنا، ونكتب إزاي نتعامل معاه (Exception) لو حصل غلط catch:

يمكن استخدامه ( جزء بيتنفذ في الآخر سواء حصل غلط أو لا :finally  
(...، connection، تقفل فايل).

## مثال عملي في C#:

```
class Program
{
    static void Main()
    {
        try
        {
            Console.Write("دخّل رقم: ");
            int num1 = Convert.ToInt32(Console.ReadLine());

            Console.Write("دخّل رقم ثاني: ");
            int num2 = Convert.ToInt32(Console.ReadLine());

            int result = num1 / num2;
            Console.WriteLine("النتيجة = " + result);
        }
    }
}
```

```
catch (DivideByZeroException)
{
    Console.WriteLine("ماينفعش نقسم على صفر يا معلم");
}
catch (FormatException)
{
    Console.WriteLine("لازم تدخل رقم صحيح مش نص");
}
catch (Exception ex)
{
    Console.WriteLine("حصل خطأ غير متوقع " + ex.Message);
}
finally
{
    Console.WriteLine("خلصنا التجربة");
}
}
```

# Custom Exceptions:

ال Exceptions الجاهزة (زي DivideByZeroException أو FormatException) بتغطي أغلب الحالات. لكن ساعات إنت في مشروعك محتاج تعرف Exception مخصوص بيمثل مشكلة عندك. مثلاً:

لو عندك نظام تسجيل دخول، ومحتاج تغطط المستخدم بـ InvalidLoginException.

أو في بنك، تعمل InsufficientBalanceException.

## إزاي نعمل Custom Exception؟

في #C أي Exception جديد بنعمله عن طريق وراثة (inheritance) من الكلاس Exception.

مثال:

```
class InsufficientBalanceException : Exception
{
    public InsufficientBalanceException(string message) :
    base(message)
    {
    }
}
```

```
class BankAccount
{
    public int Balance { get; private set; }

    public BankAccount(int initialBalance)
    {
        Balance = initialBalance;
    }

    public void Withdraw(int amount)
    {
        if (amount > Balance)
        {
            throw new InsufficientBalanceException("الرصيد مش  
الكافي للسحب");
        }
        Balance -= amount;
        Console.WriteLine($"الرصيد الحالي = {Balance} تم السحب {amount}");
    }
}
```

```
class Program
{
    static void Main()
    {
        BankAccount account = new BankAccount(100);

        try
        {
            account.Withdraw(150);
        }
        catch (InsufficientBalanceException ex)
        {
            Console.WriteLine("غلط مخصص: " + ex.Message);
        }
    }
}
```