---------- Exception Handling ----------

Exception هو استثناء او اكشن غير متوقع حدوثة في الوقت الي البرنامج يتنفذ فيه. مثل==> int Division = 5/0;

Exception Handling : هي آلية أو طريقة التعامل مع هذي الاستثنائات في الكود علشان البرنامج يتجنب ذي الأخطاء و يشتغل بدون Errors أثناء تنفيذ البرنامج.

طريقة التعامل مع الاستثنائات: المكونات الأساسية للكود المطلوب تنفيذه الخاص ب **ExceptionHandling:**

1- **Try Block** : try {} نقدر نكتب داخله الكود الي من المتوقع او ممكن يسبب استثناء.
2- **Catch Blocks:** catch {} نقدر نكتب داخله الكود الي نبغاه يعالج الاستثناء أو طريقة تعاملك انت كمبرمج مع هذا ال Error و يمكنك ان تستخدم اكثر من catch block{} لتنفيذ المطلوب .
3- **Finally Block**: finaly {} نكتب داخله الكود الي نبغاه يتنفذ اجباري بعد م يخلص من try و catch في الاغلب يحتوي على رسالة تظهرها للمستخدم تشرح فيها الية ما تم معالجته أو تنفيذ شيئ خاص بك انت كمبرمج مثل التعامل مع هذا الايرور عن طريق انك ترسل شي لقاعدة البيانات و زي كذا .

Example :
```
int x = 0;
int Division = 0;

try
{
    Division = 3 / x;
}
catch (Exception r)
{
    Console.WriteLine($"Exception is: {r.Message}");
}
finally
{
    Console.WriteLine($"Result: {Division}");
```

Output:
```
Exception is: Attempted to divide by zero.
Result: 0
```

}

 # ملاحظات:

- تقدر تستخدم try و catch بدون احتياجك ل finally block .
- في المثال السابق استخدمنا ايرور ماسج جاهزة محجوزة من كلاس Exception و التي ستظهر للمستخدم النص الآتي **Attempted to divide by zero**.
- في حالة استكمال الكود بيظهر للمستخدم **Result : 0** بعد المسج السابقة .