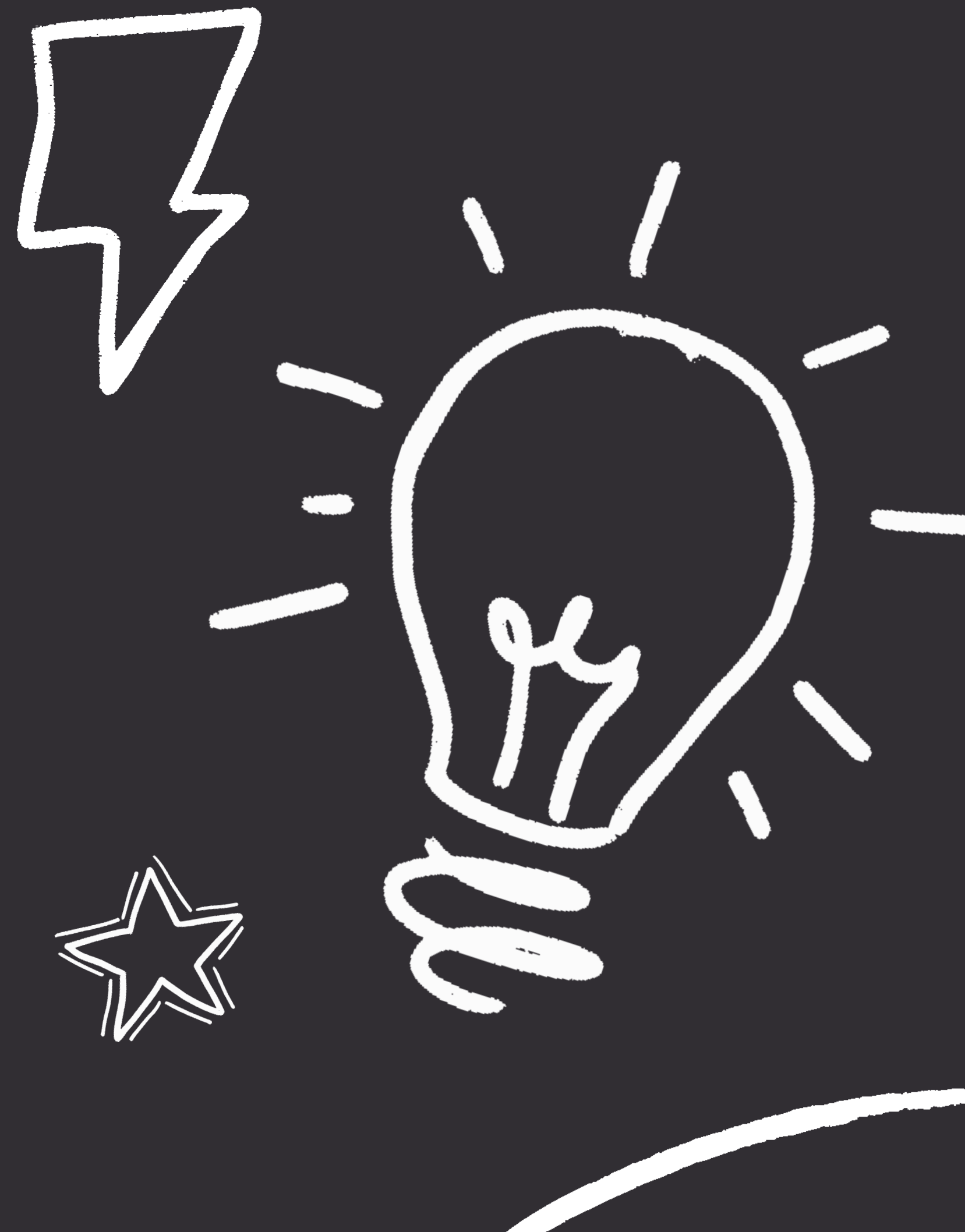


Exception Handling

Presenter :
Abdelwahab Shandy



What is Exception ?

It means an error. In the end, the program is doing something that contradicts the logic of the program

There are two types of errors in .Net

Compile Time Errors

It means an error that appears during the compile process, meaning when the program is running and discovers the error and the program is not running at all, and it is very easy to discover the error, but it is considered explicit

EX :

```
int Number = "Abdelwahab"
```

Run time Errors

EX :

```
Console.WriteLine(10 / 0);
```

Type Error :

Division by constant zero

Compile Time Errors

Runtime Errors

A logical error is detected at the time of running the program. The compiler has difficulty detecting the error because it depends on the code document at the time of execution.

How do I do Exception Handling?

The purpose of Exception Handling is to make the program handle any errors, so that when an Exception occurs, it logs it and the program continues working normally.

EX :

```
int A = 10, B = 0; int result = A/ B;           // Runtime error: DivideByZeroException
```

Runtime Error Handling:

```
try {  
int A = 10, B = 0;  
int result = A/ B;  
}  
catch ( Exception ex)  
{  
Console.WriteLine(ex.ToString());  
}
```

ex.ToString() :

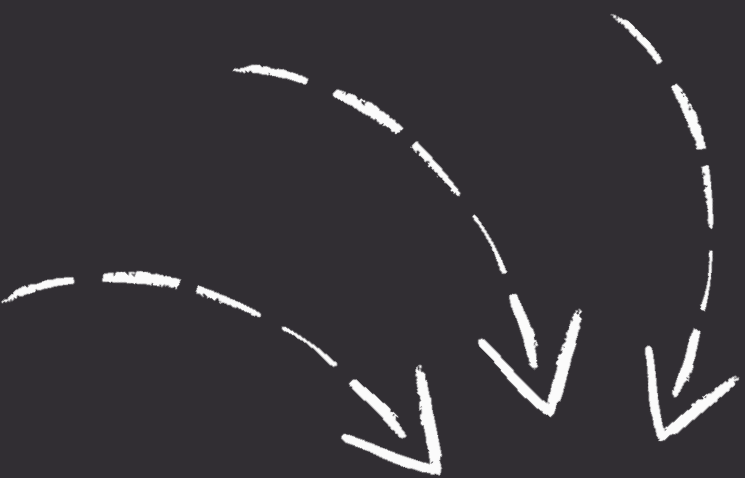
displays:

The name of the exception type.

The detailed error message

How does a Stack Trace appear?

- When an exception occurs in the program, the Stack Trace is displayed as part of the exception details.



```
System.DivideByZeroException: Attempted to divide by zero.  
   at Exception_handling__1.Program.Main(String[] args) in C:\Users\FM\Desktop\Bank Syatem\Exception handling - 1\Program.cs:line 10
```

Stack Trace Benefits:

- Helps to know where the error occurred
- Used during debugging

Important note:

- Once the execution occurs, it stops and the following lines are not executed.

C# Example

```
try {  
    int A = 10;  
    int B = 0;  
    int result = A / B ; // This will throw a DivideByZeroException  
}  
catch (Exception ex)  
{  
    Console.WriteLine(ex.ToString());  
}  
finally  
{  
    Console.WriteLine("Execution terminated.");  
}
```

try:

Contains the code that might cause an exception.

If an exception occurs in this block, execution of the code inside it stops and the program moves to the exception handler.

catch:

Used to handle the exception.

Can contain multiple catch blocks to handle different types of exceptions.

finally:

Contains the code that will be executed whether or not an exception occurs.

Typically used to perform cleanup operations such as closing files or connecting to a database.

throw

Used to throw an exception explicitly.

You can use built-in exceptions or create custom exceptions
throw new Exception("Custom error message");

```
try
{
    Console.WriteLine("Enter 5 charactar string : ");
    var input = Console.ReadLine();
    if (input.Length != 5)
        throw new Exception("string must 5 charactar long");
    Console.WriteLine(input);
}
catch (Exception ex)
{
    Console.WriteLine(ex.ToString());
}
```

```
Enter 5 charactar string :
aaaa
System.Exception: string must 5 charactar long
   at Exception_handling__1.Program.Main(String[]
args) in C:\Users\FM\Desktop\Bank Syatem\Exception
handling - 1\Program.cs:line 12

C:\Users\FM\Desktop\Bank Syatem\Exception handling
- 1\bin\Debug\net8.0\Exception handling - 1.exe (pr
ocess 30164) exited with code 0 (0x0).
Press any key to close this window . . .
```



Thank You



Abdelwahab Ahmed Shandy

abdelwahabshandy@gmail.com