

What are Exceptions?

This lesson will give you an overview of what exceptions are and their types.

WE'LL COVER THE FOLLOWING ^

- Introduction
- Explanation
 - Code Definitions for Exception Handling
 - Types

Introduction

Exception Handling is the main feature of `OOP` Concepts.

Before going to know about this Exception Handling, we should know **what is an Exception?**.

An exception is a runtime error which occurs while executing the program. Due to this our system may hang or happen some thing.

Explanation

Code that detects an error condition is said to throw an exception and code that handles the error is said to catch the exception. An exception in C# is an **object** that *encapsulates* various pieces of information about the error that occurred, such as the stack trace at the point of the exception and a descriptive error message.

All exception objects are instantiations of the `System.Exception` class or a *child class* of it.

Programmers may also define their own class inheriting from `System.Exception` or some other appropriate exception class from the .NET Framework.

Code Definitions for Exception Handling

- **try/catch** - Do something and catch an error, if it should occur.
- **try/catch/finally** - Do something and catch an error if it should occur, but always do the `finally`.
- **try/finally** - Do something, but always do the `finally`. Any exception that occurs, will be thrown after `finally`.

Types

Exceptions are caught in the order from most specific to least specific. So for example, if you try and access a file that does not exist, the CLR would look for exception handlers in the following order:

- `FileNotFoundException`
- `IOException` (base class of `FileNotFoundException`)
- `SystemException` (base class of `IOException`)
- `Exception` (base class of `SystemException`)

If the exception being thrown does not derive from or is not in the list of exceptions to catch, it is thrown up the call stack.

You've got an introduction of what exceptions are - now let's move onto their syntax and examples!