C# Basics

INSTRUCTOR: ADEEN BIN WAHEED

LECTURE 5

Mathematical functions and constants

Trigonometric functions

Sin, Cos, Tan, Asin, Acos, Atan

Argument and return types are double.

Numerical functions

Abs, Max, Min, Ceiling, Floor, Round(nearest int)

Other useful functions

Sqrt, Cbrt, Pow, Exp, Log, random

Constants

E and PI

How to use Math Function

Running Demo

Radom Numbers

```
class Program
   {
        static void Main(string[] args)
        {
              Random r = new Random();
            //Get three random numbers that will be always 5, 6, 7, 8 or 9.
            Console.WriteLine(r.Next(5, 10));
            Console.ReadKey();
        }
   }
}
```

Exception

An exception is a problem that arises during the execution of a program. When an Exception occurs the normal flow of the program is disrupted and the program/Application terminates abnormally, which is not recommended, therefore these exceptions are to be handled.

An exception can occur for many different reasons, below given are some scenarios where exception occurs.

- A user has entered invalid data.
- A file that needs to be opened cannot be found.
- A network connection has been lost in the middle of communications or the Virtual Machine has run out of memory.

Why Exception Handling?

We seek robust programs.

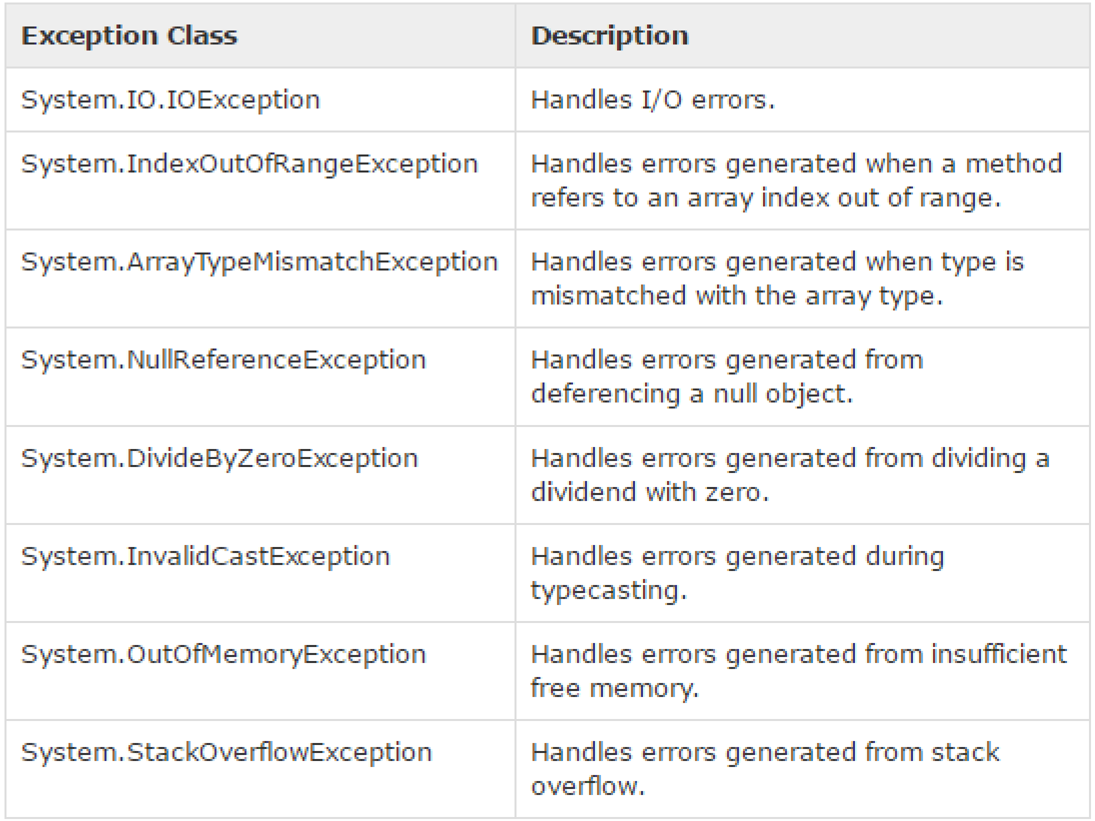
When something unexpected occur ensures that program detects the problem.

Extensive testing of special situations can result in "spaghetti code“.

Need mechanism to check for problem where it could occur.

When condition does occur should have a control passed to code for handling the problem.

Types of Exceptions in C#



Exception Class	Description
System.IO.IOException	Handles I/O errors.
System.IndexOutOfRangeException	Handles errors generated when a method refers to an array index out of range.
System.ArrayTypeMismatchException	Handles errors generated when type is mismatched with the array type.
System.NullReferenceException	Handles errors generated from deferencing a null object.
System.DivideByZeroException	Handles errors generated from dividing a dividend with zero.
System.InvalidCastException	Handles errors generated during typecasting.
System.OutOfMemoryException	Handles errors generated from insufficient free memory.
System.StackOverflowException	Handles errors generated from stack overflow.

Exception Handling

1st Way:

Code that could generate errors put in **try** blocks.

Code for error handling enclosed in a **catch** clause.

The **finally** clause always executes.

2nd Way:

Termination model of exception handling.

The block in which the exception occurs expires.

throws clause specifies exceptions method throws.

General Structure Example

Try Catch Blocks:

```
try
{ }
catch(IOException i)
{ }
```

Multiple Catch Blocks:

```
try
{ }
catch(IOException i)
{ }
catch(FileNotFoundException f)
{ }
```

General Structure Example

Throw Exception:

```
If(condition)

throw new IOException("sorry device error);
```

Exception Handling

Running Demo