C# - Exception Handling

Two types of errors can be experienced in a program:

Compile-Time errors:
- IDE/Compiler finds error; *red squiggle*
- Syntax, spelling, unreachable code, data not accessible, method parameter errors
- Code cannot be run

Run-Time errors (*aka Exceptions*)
- System finds error when code is run
- Array index out of range, out of memory, file not found, I/O error, network shutdown in middle of processing, impossible operation
- Program terminates immediately with error from system (if not handled)

Examples of Some System Exceptions Types *(there are 100's of Exceptions)*:
- **Arithmetic exceptions:** DivideByZeroException, OverflowException
- **Array exceptions:** IndexOutOfRangeException, RankException
- **Conversion exceptions:** InvalidCastException
- **IO exceptions:** IOException, FileNotFoundException, DirectoryNotFoundException
- **Null reference exceptions:** NullReferenceException
- **Argument exceptions:** FormatException ArgumentNullException, ArgumentOutOfRangeException
- **Type exceptions:** InvalidCastException, TypeLoadException
- **Threading exceptions:** ThreadAbortException, ThreadStateException
- **Serialization exceptions:** SerializationException
- **Database exceptions:** SQLException (for ADO.NET)

To handle *(react or process)* an Exception, use a **try / catch** block.

```
try
  {
     // Code that might throw an exception
  }
catch(ExceptionType exceptionVariable)
     {
        // Code to handle the exception
     }
```

If the code within the **try** block causes an Exception, C# looks for a **catch** block following the **try** block for the Exception. Any code following the statement causing the exception is skipped.

If there is **catch** block for the Exception, the code in the **catch** block executes and the program continues at the end of the **catch** block (or all **catch** blocks if multiple **catch** blocks are coded)

If there is no **catch** block for the Exception, the program terminate.

An Exception block is passed to the `catch` block using name you specified in the catch block

The Exception block contains information abut the exception you can use, if you want to.

Exceptions "***bubble up***" to the most recent catch block that handles the exception.