# Getting Started with Exceptions



Jason Roberts
.NET Developer

@robertsjason

dontcodetired.com

### Overview



#### Understanding exception handling

- Exception "bubbling"
- try...catch...finally

Demo code overview

Causing an exception

Understanding the stack trace

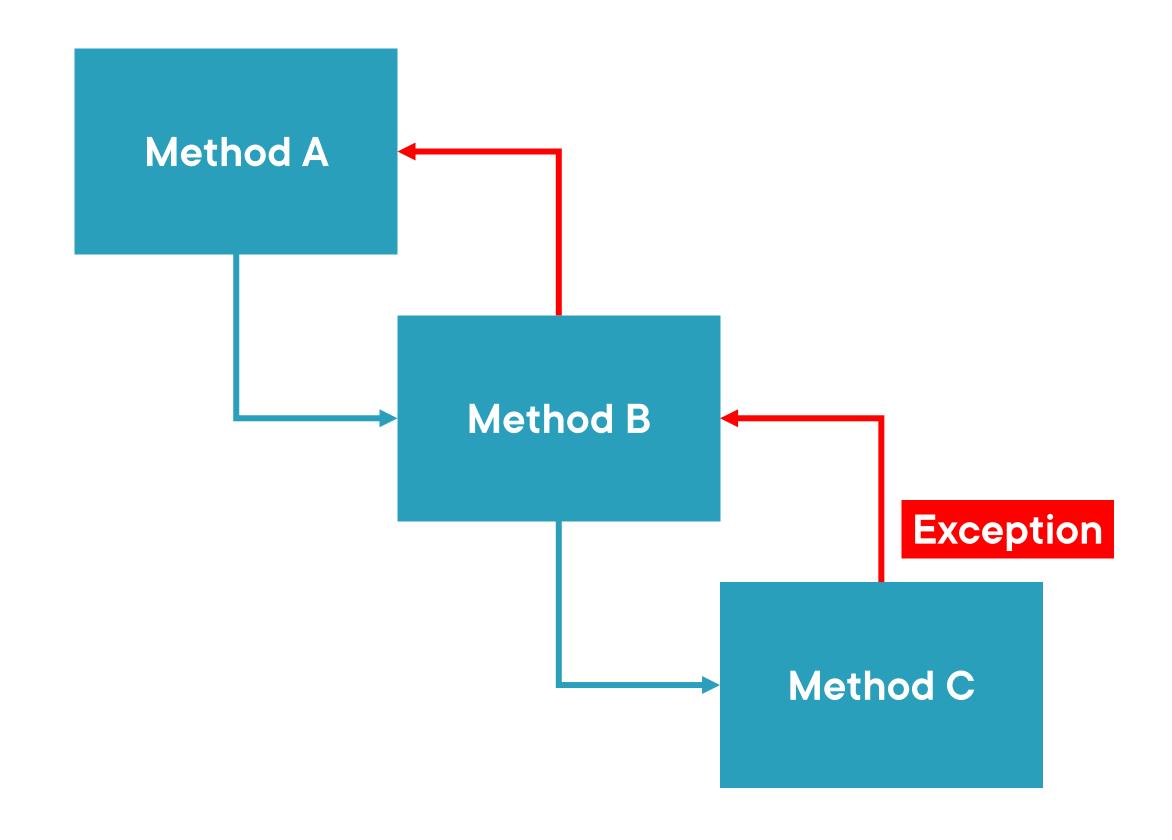
Creating and throwing an exception

Getting started with exception catching

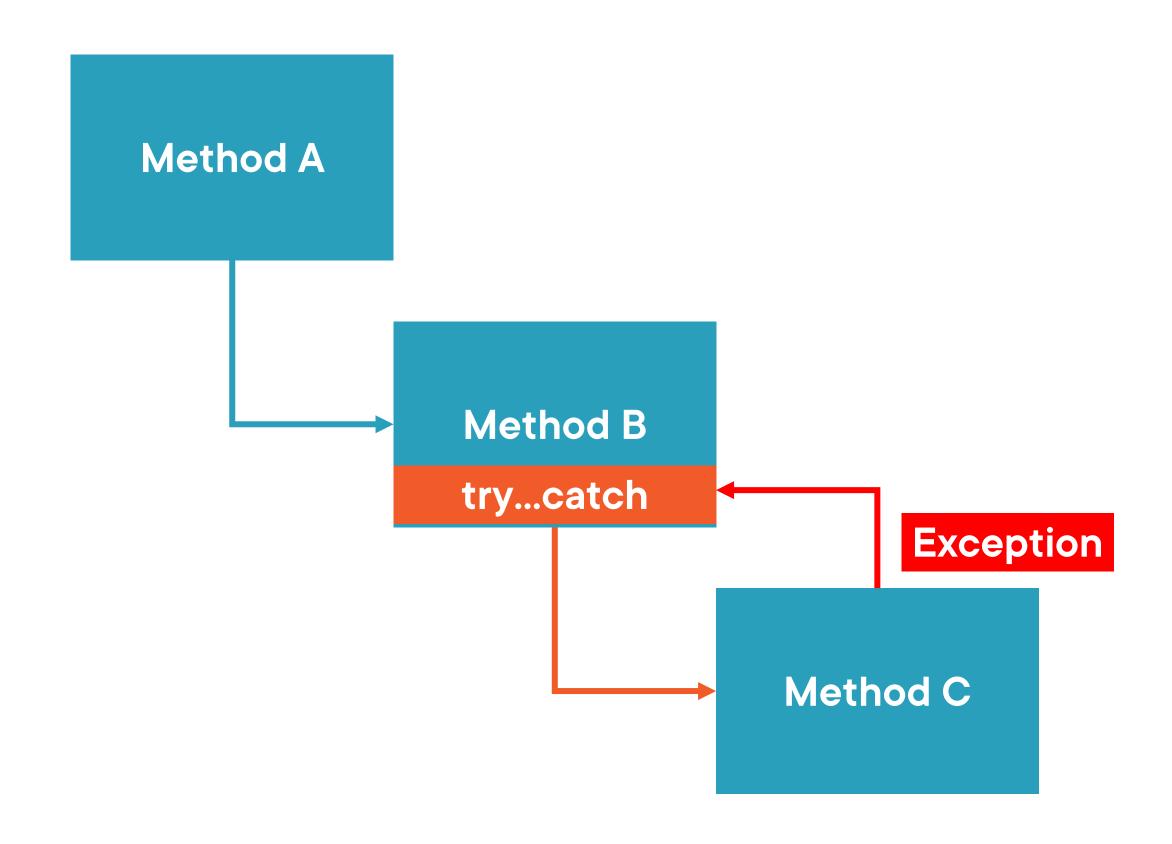
**Exception handling good practices** 



# Understanding Exception Handling



## Understanding Exception Handling



```
try
    // Some operation(s)
catch (ArgumentNullException ex)
    // Handle ArgumentNullException
catch (InvalidOperationException ex)
    // Handle InvalidOperationException
catch (Exception ex)
    // Handle all other exceptions
```

Most specific

Least specific



```
try
    // Some operation(s)
catch (ArgumentNullException ex)
    // Handle ArgumentNullException
catch (InvalidOperationException ex)
    // Handle InvalidOperationException
catch<
    // No exception variable
```



```
try
{
    // Some operation(s)
}

finally
{
    // Always executed when control leaves try block
}
```



```
try
    // Some operation(s)
catch (ArgumentNullException ex)
    // Handle ArgumentNullException
finally
    // Always executed when control leaves try block
```



Do not add a catch block that does nothing or just rethrows

Catch block should add some value

May just be to log the error

Usually bad practice to ignore (swallow/trap) exceptions



Do not use exceptions for normal program flow logic

#### E.g. input validation

- You expect input to be invalid sometimes
- Not an exceptional situation
- Part of expected logic flow

IsValid(xxx) method(s)



Design code to avoid exceptions

```
int Parse(string input)
bool TryParse(string input, out int result)
if (cn.State != ConnectionState.Closed)
 cn.Close();
Consider returning null (or null object
pattern) for extremely common errors
```

Use correct grammar in exception messages

**Correct punctuation** 

**Correct spelling** 

End sentences with full stop

Consider error message localization



Use finally blocks for clean-up

E.g. calling Dispose()

Callers of methods should be able to assume no unexpected side effects when exception thrown/caught



### Summary



#### Understanding exception handling

- Exception "bubbling"
- try...catch...finally

#### Caused a DivideByZeroException

- Visual Studio debugger
- Windows event viewer

Stack trace (and other exception properties)

Threw ArgumentOutOfRangeException

**Catching exceptions** 

**Exception handling good practices** 



# Up Next:

Understanding the Exception Class Hierarchy

