Tasks

## Coding Practice

For threading programming questions see

linqpad\Queries\InterviewQuestions\Threading\Tasks

## Overview

Questions

What do tasks do?

Separates specification and co-ordination of units of work from the scheduling details

Why use tasks?

1. Separate the specification and co-ordination of units of work from their scheduling.
2. Easy to get a return value from a task.
3. Tasks can be chained together
4. Exceptions throws by unit of work re-thrown when Wait/Result called
5. Standardised cancellation protocol

What happens if one does not handle unobserved exception on autonomous tasks?

They fail silently.

How can one obtain unobserved exceptions?

By registering TaskScheduler.UnobservedTaskException

TaskScheduler.UnobservedTaskException += (source,args) => {

Console.WriteLine(args.Exception);

};