



Presentation

- Namn: Mattias Andersen
- Programmerat konstant de senaste 4 åren
 - Bor med fru och två barn
- Sysslar även mycket med musik (basist)
 - Webbutvecklare på Edument AB

Är en av två utvecklare på Decemberborn som just nu skapar “Cathedral”



Tekniker/språk: LUA-script, C++,
Python och en del pixel art :)

Arbetat i stora projekt som kört:

- React
- Angularjs
- Git (I massor)
- TDD (givetvis)

Avbryt mig med era frågor!

- Även frågor som inte rör C#/.NET är välkomna!

FEEDBACK!

Skicka gärna era synpunkter på kursmaterialet:

Tentamen

Tentamen är skriftlig och kommer att baseras på de delar som vi tar upp i kursen.

Obligatoriska uppgifter

Vi kommer att köra några obligatoriska uppgifter under denna kurs.

Det är 6st som ska göras och de finns som inlämningsuppgifter i Ping pong.

Även ett projekt som görs i par eller grupper om 3 personer är obligatoriskt.

Redovisning av uppgifter

- Allt ska pushas till github.
- **En-och-en** redovisar för läraren.
(Eftermiddagar)
- När ni är godkända skickar ni in en tom fil till rätt uppgift i ping pong så vi kan bocka av.
- Blir ni icke-godkända får ni komplettera.
- Allt ska vara klart **innan första tentan.**

Kört fast?

- Kör ni fast, fråga då först och främst era studiekamrater. Det är bra att hjälpa varandra!
- Försök Googla eller läs på! För nu är fokus på att ni måste lösa problemen på egen hand, så som det är i verkligheten!

Visual Studio

- Visual Studio 2015 eller Visual Studio 2017

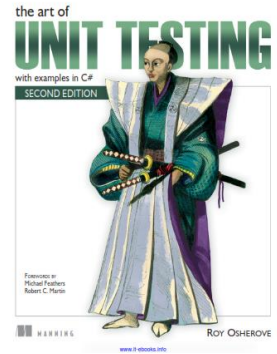
Test-framework

- NUnit 3.x

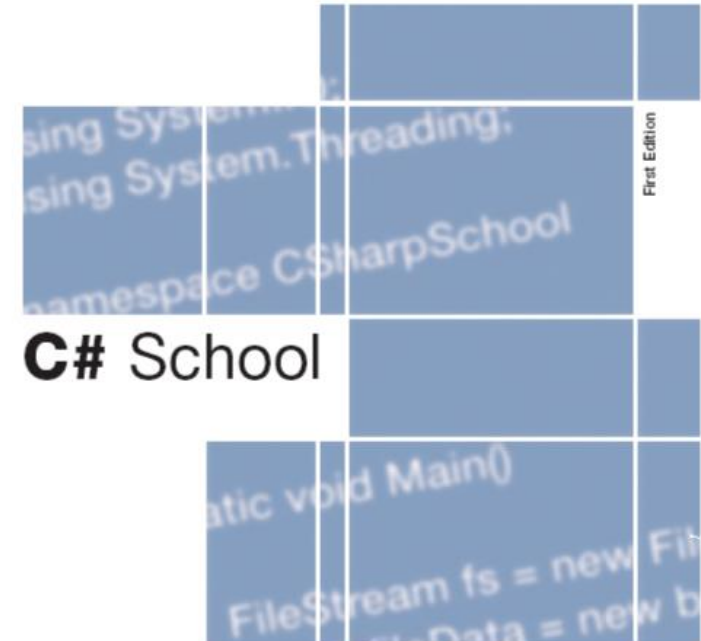
Mocking-ramverk

- NSubstitute

- **Kurslitteratur:**
- **The Art of Unit Testing, 2nd edition**
Roy Osherove
ISBN: 1617290890



- **C# School**
- A free e-book that we published in 2006
- Gives you 14 chapters on getting started with C# and .NET.
- Written by Tore Nestenius, Jonathan Worthington and more.



14 lessons to get you started with C# and .NET

Author: Faraz Rasheed

Editors: Tore Nestenius
Jonathan Worthington
Lee Addy

www.programmersheaven.com

http://www.programmersheaven.com/ebooks/csharp_ebook.pdf

- **Kursens huvudsakliga innehåll**

Kursens huvudsyfte är att ge deltagaren fördjupad teoretisk och praktisk kunskap om TDD (Test Driven Development).

- Den studerande lär sig att skriva tester till sin kod innan själva koden är skapad. Det innebär att man undviker de dyra logiska felen som man annars inte ser förrän koden kommit ut i produktion.
- Kursen kommer att gå genom de tekniska delarna som krävs för att kunna skriva TDD. Det inkluderar olika ramverk för test, olika program och code-patterns med best practices.

Kursen omfattar följande moment enligt kursplanen:

- Unit testning med NUnit och MSTest från Visual Studio Test System (VSTS)
- Testdrivers – NUnit GUI, TestDriven.Net, VSTS
- Mocks och stubbar
- TDD – Test Driven Development
- DDT – Design Driven Testing
- CodeCoverage – NCover
- Dependency Injection (DI)
- C# kod-stil och designregler

Målet med kursen:

Efter kursen ska studenten kunna:

- Förstå och kunna skriva testbar och läsbar kod med applicering av TDD och dess metoder.
- Få en inblick i de vanligaste verktygen som används inom testning.

Vi tillämpar traditionell undervisning i sal, med gemensamma föreläsningar och praktiska laborationer.

Föreläsningarna är jämnt utspridda över hela kursen och laborationerna görs enskilt eller i grupp, allt efter utbildarens direktiv.

I kursen ingår också att slutföra de obligatoriska uppgifterna under självstudietiden.

Laborationerna redovisas muntligt under angivna laborationstider till utbildaren eller via annan överenskommelse och måste godkännas av utbildaren med betyget G. För tentamen gäller betygen nedan.

För examination gäller betyget G på samtliga laborationer sammanvägt med resultat på sluttentamen.

För att få betyget **godkänt** krävs att den studerande

- har erhållit lägst betyget godkänt på tentamen
- har klarat av alla obligatoriska moment
- obehindrat kan skriva Asserts till MSTEST och NUnit
- förstår vikten av att kunna skriva i TDD när man vill skapa felfri kod
- kan redogöra för TDD-processen

För att få betyget **väl godkänt** krävs att den studerande

- uppnått kunskapskraven för betyget godkänt
- har erhållit lägst betyget väl godkänt på tentamen
- förstår hur man arbetar med dependency injection för att skapa lösa kopplingar
- kan skapa Mock objekt och stubbar som simulerar resurser likt databaser

Former för kunskapskontroll

- Redovisning av obligatoriska moment (muntligt/skriftligt)
 - Individuell skriftlig tentamen