

# Learn to code like a "Craftsman"

The Software Craftsmanship movement strives to raise the bar for professional software development by practicing it and helping others to learn the craft. In this course we want to inspire you by practicing with us and helping us learn together.

**DESCRIPTION.** In our brave new agile world where cross functional teams work together in parallel to add features in huge code bases, new demands are put on us software developers. Getting your program to work according to specification is no longer good enough. The code must be well-written and well-tested so that it can be easily understood and safely modified by the next developer.

The software craftsmanship manifesto addresses this need by emphasizing well-crafted software, steadily adding value to the software and working together as a community of professionals.

In this course we will practice writing what we call "Clean code" by the means of pair/group programming and Test Driven Development. We will discuss program structure, naming, comments, SOLID principles, refactoring and more. While the course will include theory segments, the focus will be on programming in practice where we hope you will learn from others as well as letting others learn from you.

Since this course is not about any specific programming language or tool, everyone is encouraged to bring their own programming environment. We will of course supply laptops with various programming environments to those who do not bring their own.



## TARGET

### AUDIENCE.

Software developers in all stages of their career.

**PREREQUISITES.** You should be fairly confident with at least one programming language.

**COURSE GOAL.** By the end of this course we hope you will be inspired to continue practicing at home, to seek out or start up local communities and continue to grow as a software craftsman.

**DURATION.** 2 days

**COURSE RESPONSIBLE.** Micael Andersson

**COURSE BOOK.** Robert C. Martin's "Clean Code" plus handouts

**LANGUAGE.** English

**COURSE CATEGORY.** System development

## CONTENTS

- Introduction to Software craftsmanship
- Clean code
- Practicing by doing katas
- Pair programming/randoris
- TDD, including refactoring and mocking