

THE QUANTS HUB
PROGRAMMING SCHOOL
PRESENTS



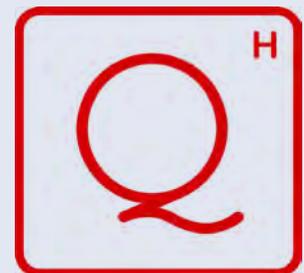
F# AND FUNCTIONAL PROGRAMMING IN FINANCE BY TOMAS PETRICEK 10 WEEK ONLINE COURSE

Functional-first programming in F# is an effective tool for solving complex problems that often arise in financial computing. The strong typing of F# provides important correctness guarantees and means that numeric code written in F# runs efficiently.

THE COURSE REQUIRES NO PRIOR FUNCTIONAL PROGRAMMING KNOWLEDGE AND IS DESIGNED FOR BOTH SOFTWARE DEVELOPERS AND QUANTS OR ACTUARIES.

YOU WILL LEARN

- HOW TO APPROACH PROBLEMS FROM THE FUNCTIONAL PERSPECTIVE AND CAPTURE YOUR IDEAS USING IDIOMATIC F#.
- MODEL A PROBLEM DOMAIN, SUCH AS STOCK OPTIONS, USING FUNCTIONAL TYPES AND DEVELOP DOMAIN SPECIFIC LANGUAGES (DSLs) FOR PROCESSING SUCH DOMAINS.
- USE TYPE PROVIDERS TO ACCESS DATA, PERFORM INTERACTIVE DATA AND TIME-SERIES ANALYSIS ON FINANCIAL DATA USING THE DEEDLE LIBRARY AND CREATE CHARTS TO VISUALIZE THE RESULTS.
- HOW TO USE F# WITHIN A LARGER CONTEXT, INCLUDING INTEROPERABILITY WITH R AND BEST PRACTICES FOR THE ENCAPSULATION OF F# COMPONENTS FOR .NET.



Quants Hub

COURSE START DATE:
5TH OCTOBER 2015

CERTIFIED BY



COURSE OVERVIEW

```
// NOTE ABOUT the line Seq.reduce addNumbers {0L..10000000L}
// Defines a function that will add all the numbers from 0 to 10,000,000
// Uses Seq.reduce function that takes a function as the first parameter and list to apply that fu
// The first parameter is the addNumbers function.
// The second parameter is a sequence of numbers 0-> 10,000,000.
// In F# a sequence is an alias for IEnumerable<T>
// (only one object is in memory in a given time, as this is an implementation of the iterator
// Note: In F# you can easily create a range sequence by specifying a head and tail separated by "
{0L..10000000L} creates an iterator construct that will "yield" one member at a time
// Note: A list range can be created as such..[0L..10000000L].
// This is equivalent to List<int64> in C#.
// Lists are created in memory, therefore all of the members are created in memory before the
// Lists (for obvious reasons) can cause memory exceptions when working with large data sets
```

F# AND FUNCTIONAL PROGRAMMING IN FINANCE WITH TOMAS PETRICEK

Each lecture will appear in your members area weekly, and all webinar invites will be sent directly. You can also contact Tomas at any time with the F# forum. We can offer additional Webinars to candidates who live in different time zones.

Functional-first programming in F# is an effective tool for solving complex problems that often arise in financial computing. The strong typing of F# provides important correctness guarantees and means that numeric code written in F# runs efficiently. Furthermore, a number of case-studies show that F# significantly reduces time-to-market, especially in the financial domain.

This course is a practical introduction to the F# language, functional programming and their use in the financial domain. You'll learn about F#, fundamental functional concepts as well as libraries for numeric computing, data analysis. You'll also become familiar with best practices for using F# tools and integrating F# with larger systems.

The course is practically focused. Throughout the course, we look at examples of time-series analysis, modelling and pricing of stock options and more. Each lecture provides a number of fun exercises that guide you through the problem. Furthermore, F# and functional programming makes you a better programmer even if you do not end up using the language immediately after the course.

ABOUT THE PRESENTER



Tomas Petricek is a long-term F# enthusiast, frequent conference speaker and an author of "Real-World Functional Programming".

He is a founder of DualNotion Ltd. where he provides training and consulting services.

Tomas contributed to the development of F# as a contractor at Microsoft Research, authored Try F# tutorials on financial computing and recently spent 3 months in New York, working on financial data analytics tools for F# at BlueMountain Capital.

COURSE PREREQUISITES

To take the course programming experience is expected in some languages; C, VB, Fortran, Matlab etc. Experience with C, C++ will also come in useful for some topics. Object oriented programming skills are not totally necessary but will also help. A knowledge of calculus, statistics, signal and image processing, optimization will all assist your learning but are not absolutely required.

COURSE FEE

The regular course fee is **£799.00**.



PRMIA CERTIFICATION

The Professional Risk Managers' International Association (PRMIA) is a non-profit professional association, governed by a Board of Directors directly elected by its global membership, of nearly 90,000 members worldwide. PRMIA is represented globally by over 65 chapters in major cities around the world, led by Regional Directors appointed by PRMIA's Board. For more information visit: <http://www.prmia.org>



CPD CERTIFICATION

You will be able to receive up to 27 CPD points (9 hours of structured CPD and 18 hours of self-directed CPD) for completing this course.

The CPD Certification Service was established in 1996 as the independent CPD accreditation institution operating across industry sectors to complement the CPD policies of professional and academic bodies. The CPD Certification Service provides recognised independent CPD accreditation compatible with global CPD principles. www.cpd.uk.co.uk

CONTACT:

Tel: +44 (0)1273 201 352 / Fax: +44 (0)1273 201 360

www.quantshub.com / sales@quantshub.com

10 WEEK COURSE SCHEDULE

WEEK 1. LECTURE 1. INTRODUCING F# AND FUNCTIONAL PROGRAMMING

We quickly look at the main reasons for adopting F#. Why is it becoming popular in the finance industry and what are some successful case studies? Then we introduce the fundamental F# language features such as immutability, tuples and pattern matching.

WEEK 2. LECTURE 2. WORKING WITH COLLECTIONS AND DATA STRUCTURES

This lecture introduces the most important functional pattern – processing of immutable data structures using higher-order functions. We finish the processing of historical stock prices from Yahoo! Finance, calculating statistics and visualizes the result with simple charts.

WEEK 3. LECTURE 3. IMPLEMENTING MATHEMATICAL CALCULATIONS

F# makes it easy to turn mathematical equations to code. In this lecture we look at examples such as Monte-Carlo simulations, Black-Scholes equation and calculating historical volatility. You'll learn how to avoid mistakes with units of measure, how to write efficient numerical code and how to use the rich Math.NET library.

WEEK 4. PRACTICAL EXERCISE & WEBINAR WEEK

This will cover the first 3 weeks of the course. The practical exercise will be marked and feedback given.

WEEK 5. LECTURE 4. DOMAIN SPECIFIC LANGUAGES FOR FINANCE

Domain specific languages (DSLs) are an effective way of solving recurring problems. In this lecture, we build a DSL for pricing financial options and for detecting patterns in changing prices. You'll learn how to model problem domain using functional data structures and how to build an easy to use library on top of the model.

WEEK 6. LECTURE 5. EXPLORATIVE DATA AND TIME-SERIES ANALYSIS

In this lecture we look at F# type providers and Deedle. Type providers make it easy to access data from sources including CSV and XML files, Excel, SQL databases and Web and REST services. Using Deedle we can then align multiple time-series and perform interactive analysis – such as comparing different industry sectors or calculating daily returns.

WEEK 7. LECTURE 6. F# IN THE LARGER CONTEXT

We wrap up by looking at the ways for integrating F# in the broader context. This lecture explores how to call advanced statistical libraries using the R provider, how to use object-oriented programming to integrate with .NET and how to use F# tools and libraries for unit testing, building and documenting code.

WEEK 8. PRACTICAL EXERCISE & WEBINAR WEEK

This will cover weeks 5-7 of the course. The practical exercise will be marked and feedback given.

WEEK 9. REVISION WEEK.

WEEK 10. FINAL PRACTICAL PROJECT WEEK.

THE FINAL PROJECT WILL BE MARKED WITH FEEDBACK AND A PASS OR FAIL WILL GIVEN. ONE RETAKE IS ALLOWED IF YOU FAIL.

THIS COURSE IS AVAILABLE GLOBALLY AND CAN ONLY BE ACCESSED ONLINE.

BOOK YOUR PLACE NOW: [HTTP://QUANTSHUB.COM/CONTENT/F-AND-FUNCTIONAL-PROGRAMMING-FINANCE](http://quantshub.com/content/f-and-functional-programming-finance)



**F# AND FUNCTIONAL PROGRAMMING IN FINANCE
10 WEEK ONLINE COURSE
5TH OCTOBER 2015**

COURSE FEE STRUCTURE

Regular Course Fee

Online Workshop:

£799.00 inc. UK VAT

Special Discount Code:

50% Academic Discount / FULL-TIME Students Only

DELEGATE DETAILS
COMPANY:
NAME:
JOB TITLE/POSITION:
NAME:
JOB TITLE/POSITION:
NAME:
JOB TITLE/POSITION:
DEPARTMENT:
ADDRESS:
COUNTRY:
TELEPHONE:
E-MAIL:
DATE:
SIGNATURE:

TO REGISTER, PLEASE FAX THE COMPLETED BOOKING FORM TO:

+44 (0)1273 201 360

SPONSORSHIP:

Quants Hub offer sponsorship opportunities for all workshop, e-mail headers and the website. Contact sponsorship via telephone on: +44 (0)1273 201 352

DISCLAIMER:

Quants Hub command the right to cancel or alter any part of this programme.

CANCELLATION:

By completing this form, the client hereby enters into a agreement stating that if a cancellation is made by fax or writing within two weeks of the event date no refund shall be given. However in certain circumstances a credit note may be issued for future events. Prior to the two week deadline, cancellations are subject to a fee of 25% of the overall course cost.

REGISTRATION:

Tel: +44 (0)1273 201 352 / Fax: +44 (0)1273 201 360

www.quantshub.com / sales@quantshub.com