KOENIG SOLUTIONS · COURSE OUTLINE

Programming with F#

Learn to build .NET applications with F#, from core syntax and functions to collections, object-oriented features and asynchronous programming.

12	33	12
MODULES	TOPICS	HANDS-ON ACTIVITIES

COURSE OVERVIEW

This course covers F# and its place in the .NET ecosystem, along with the fundamentals of functional programming, core syntax, types, and functions including lambda expressions, pipelines, and recursion. It then addresses conditional expressions, loops, type inference, special types, and exception handling using try...with and try...finally. Further modules cover collection types, records and discriminated unions, structs, and object-oriented concepts such as classes, interfaces, and inheritance. The course concludes with computation expressions, asynchronous programming, code organisation using namespaces and modules, and interoperability with other .NET languages alongside reflection.

WHO THIS COURSE IS FOR

- **Software Developers**
- **.NET Developers**
- **Application Architects**
- **Functional Programming Enthusiasts**
- **Backend Developers**

LEARNING OBJECTIVES

- Explain the fundamentals of functional programming and F#'s role in .NET
- Apply core F# syntax including literals, bindings, and type inference
- Implement functions, lambda expressions, pipelines, and recursion
- Apply conditional expressions and loop constructs in F# programs
- Handle exceptions using try...with, try...finally, raise, and reraise
- Work with collection types such as lists, arrays, and sequences
- Implement records, discriminated unions, structs, and OOP concepts in F#
- Develop asynchronous and interoperable F# applications

COURSE SYLLABUS

12 Modules, 33 Topics

01

Introduction to F# and Functional Programming

F# is a functional-first programming language that runs on .NET, providing a unique combination of functional, imperative, and object-oriented programming. In this module, you'll get an overview of F#, learn about its place in the .NET ecosystem, and understand the fundamentals of functional programming.

TOPICS

Overview of F# and .NET · Basics of Functional Programming · Getting Started with F# in Visual Studio

02

Activity

Core F# Syntax and Types

In this module, you'll dive deeper into the syntax and types of F#. You'll learn how to write basic expressions and work with core language constructs that enable you to start developing applications.

TOPICS

Literals, Strings, and Values · Immutable Data and Type Inference · Let and Do Bindings

03

2 Activities

Working with Functions

Functions are fundamental to F# and play a pivotal role in functional programming. In this module, you'll explore the intricacies of defining, using, and optimizing functions, along with leveraging the robust pattern matching capabilities of F#.

TOPICS

Understanding Functions · Exploring Lambda Expressions and Pipelines · Recursive and Inline Functions

04

Activity

Loops and Conditions

This module focuses on exploring essential loop and conditional constructs in F#, providing foundational skills necessary for developing robust applications. You will delve into conditional expressions using 'if...then...else' statements, learn to iterate efficiently with 'for' loops for dynamic control flow in your programs.

TOPICS

Conditional Expressions (if...then...else) · Loops: for...Expression · **Loops:** while...do Expressions

05

Activity

Types and Inference

In this module we will learn fundamental types and the type inference capabilities in F#, crucial for developing maintainable codebases. You will explore the intricacies of type definitions, leveraging F#'s ability to deduce types implicitly, and delve into specialized data structures like tuples, options, and results.

TOPICS

Types and Type Inference · **Special Types:** Tuples, Options, and Results

06

Activity

Exception Handling

In this module we will learn exception handling techniques in F#, essential for writing robust and resilient applications. You will learn about different types of exceptions, how to handle them using the 'try...with' and 'try...finally' expressions, and effectively manage exceptions with 'raise' and 'reraise' functions.

TOPICS

Exploring Types of Exceptions · Implementing the try...with Expression · Applying the try...finally Expression · Utilizing Raise and Reraise Functions

07

Activity

F# Collection Types

Module 7 provides an in-depth introduction to diverse collection types in F#, essential for managing and manipulating data effectively within functional programming paradigms. You will explore the characteristics and applications of lists, arrays, sequences, slices, and reference cells.

TOPICS

Exploring Lists and Arrays · Understanding Sequences and Slices · Utilizing Reference Cells

08

Activity

Records and Unions

This module delves into advanced data structures such as Records and Unions in F#, crucial for modeling complex data and enhancing code clarity and flexibility. You will explore the intricacies of defining and utilizing Records and Discriminated Unions to represent structured data and manage variant scenarios effectively.

TOPICS

Exploring Records in F# · Mastering Discriminated Unions

09

Activity

Structs and Object-Oriented Programming

Module 9 introduces Structs and essential Object-Oriented Programming (OOP) concepts in F#, enhancing your ability to design and build modular and scalable applications. You will explore the fundamentals of Structs, Classes, Interfaces, Abstract Classes, and Inheritance within the context of functional programming.

TOPICS

Exploring Structs, Classes, and Interfaces · Mastering Abstract Classes · Understanding Inheritance in F#

10

Activity

Computation and Asynchronous Programming

Module 10 explores Computation Expressions and Asynchronous Programming in F#, pivotal for managing computations and handling asynchronous operations efficiently within functional programming paradigms.

TOPICS

Exploring Computation Expressions · Utilizing Tasks and Lazy Expressions

11

Activity

Organizing Code

Module 11 emphasizes the importance of structuring and managing code effectively using namespaces, modules, and access control mechanisms in F#. You will learn essential techniques for organizing and enhancing code clarity and reusability.

TOPICS

- Exploring Namespaces and Modules
- Implementing Access Control, Open Declarations, and Signature Files

12

Activity

Interoperability and Reflection

Module 12 explores Interoperability with other .NET languages and Reflection capabilities in F#, essential for integrating with existing codebases and accessing metadata dynamically.

TOPICS

- Exploring Interoperability with Other Languages
- Utilizing Attributes and Code Quotations
- Leveraging Caller Information and Path Identifiers

HANDS-ON ACTIVITIES

Practise What You Learn

#	MODULE	ACTIVITY
01	**Introduction to F# and Functional Programming**	**LAB 1: SETTING UP YOUR DEVELOPMENT ENVIRONMENT** ■ Install Visual Studio and F# tools ■ Write and run your first F# program
02	**Core F# Syntax and Types**	**Lab 2: Exploring F# Data Types:** Work with F# expressions and basic data types
03	**Working with Functions**	**LAB 3: IMPLEMENTING FUNCTIONAL LOGIC** ■ Practice writing and utilizing functions in F# ■ Implement higher-order functions for enhanced functionality
04	**Loops and Conditions**	**Lab 4: Building Robust Applications with Loops and Conditions:** Apply conditional expressions and various loop types to solve practical programming challenges effectively
05	**Types and Inference**	**Lab 5: Data Processing in F#:** Apply various F# types to practical data processing scenarios
06	**Exception Handling**	**Lab 6: Handling Exceptions in F#:** Practice implementing exception handling strategies in F# code
07	**F# Collection Types**	**Lab 7: Working with Collections in F#:** Apply F# collection types to real-world scenarios to enhance data handling capabilities
08	**Records and Unions**	**Lab 8: Working with Records and Unions in F#:** Implement Records and Unions in practical F# applications to handle diverse data structures
09	**Structs and Object-Oriented Programming**	**Lab 9: Object-Oriented Programming in F#:** Apply OOP principles to develop robust and extensible F# programs
10	**Computation and Asynchronous Programming**	**Lab 10: Asynchronous Programming in F#:** Apply advanced asynchronous programming techniques in F# applications
11	**Organizing Code**	**Lab 11: Organizing F# Code:** Practice organizing F# code using namespaces, modules, and access control features
12	**Interoperability and Reflection**	**Lab 12: Interoperability and Reflection in F#:** Implement Interoperability and Reflection techniques in practical F# applications

WHAT IS INCLUDED

Everything You Need to Succeed

- Koenig course material for every module
- Delivery by certified instructors and subject matter experts
- Hands-on exercises throughout the course
- Course completion certificate from Koenig

ABOUT KOENIG

Who We Are

Established in 1993, Koenig Solutions is a globally recognised training provider with 35+ technology partnerships, 5,000+ training programmes and a network of certified instructors and subject matter experts, supporting individuals and enterprise teams worldwide.

FLEXIBLE DELIVERY

Learn the Way That Suits You

Classroom Training | Live Online Classes (ILO) | Fly-Me-A-Trainer (FMAT) | Flexi (Self-Paced) | 1-on-1 Training

Ready to Get Started?

Talk to a learning advisor about schedules, group training and customised delivery.

Visit koenig-solutions.com | info@koenig-solutions.com