# Unit 1 - Introduction to Functional Programming

1. Stateful vs stateless computation
2. Lambda calculus semantics.
   1. Variables
   2. Lambda-abstractions/functions
   3. Function application
3. Introduction to F#, basic functionalities (main, print, …)
4. If-then-else, bindings, recursive bindings.
5. Examples and Exercises.

# Unit 2 – Types in Functional Programming

1. Typed lambda calculus. Typing variables, lambda abstractions, function applications.
2. Lists and recursion.
3. Structural equality vs reference equality.
4. Exercises on lists.

# Unit 3 – Higher-order Functions

1. Curry, uncurry, map, fold, map2, fold2.
2. Case Study: SQL

# Unit 4 – Data Structures

1. Tuples and Unions in lambda-calculus (implementation shown in F#).
2. Tuples, Unions, and Records in F#.
3. Pattern matching,
4. Case Study: Expression evaluation

# Unit 5 – Advanced Data Structures

1. Recursive data structures (Trees, Lists with Cons and Empty, …)
2. Function records.
3. Case Study: Imperative Language with AST.

# Unit 6 – Drawing in Functional Programming

1. Function composition
2. Drawing lines
3. Drawing shapes
4. Using function composition for rendering.

# Unit 1 - Introduction to Functional Programming