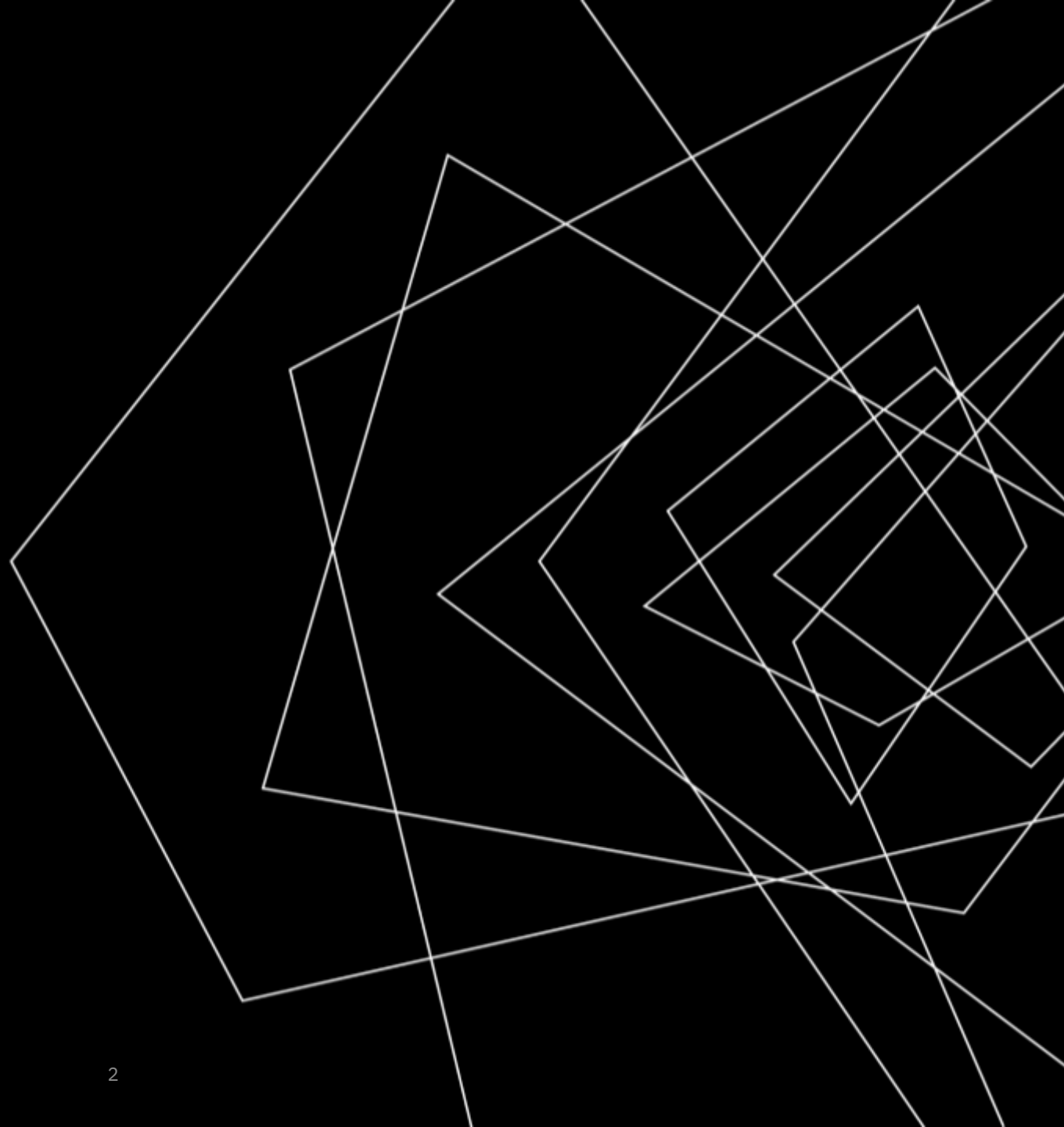
Introduction to Functional Programming and Type Theory

Mridul Gupta

Apoorv Tandon

INTRODUCTION

Functional programming is a programming paradigm in which we try to bind everything in pure mathematical functions style. It is a declarative type of programming style. Its main focus is on “what to solve” in contrast to an imperative style where the main focus is “how to solve”. It uses expressions instead of statements. An expression is evaluated to produce a value whereas a statement is executed to assign variables.



CONTENTS

TOPICS

WEEK 1

Introduction to OCaml and functional programming concepts.

WEEK 2

Introduction to fundamental type theory concepts and hands on with OCaml.

WEEK 3

Advanced OCaml and Type Systems

WEEK 4

Type Inference and Type Checking

WEEK 5

Advanced type theory topics such as dependent types and type constructors.

WEEK 6

Introduction to functional programming paradigms:
recursion, higher-order functions, and purity.

WEEK 7

Writing concurrent programs in OCaml using
different concurrency models.

WEEK 1

Introduction to OCaml and functional programming concepts

Overview of type theory and its significance.

Setting up development environment: Installing OCaml compiler, editor, and necessary libraries.

WEEK 2

Learning basic syntax and constructs in OCaml.

Introduction to fundamental type theory concepts such as type inference and polymorphism

Hands-on exercises in OCaml focusing on type annotations and polymorphic functions

WEEK 3-4

Deep dive into type systems, covering algebraic data types and parametric polymorphism

Understanding type inference algorithms and type checking

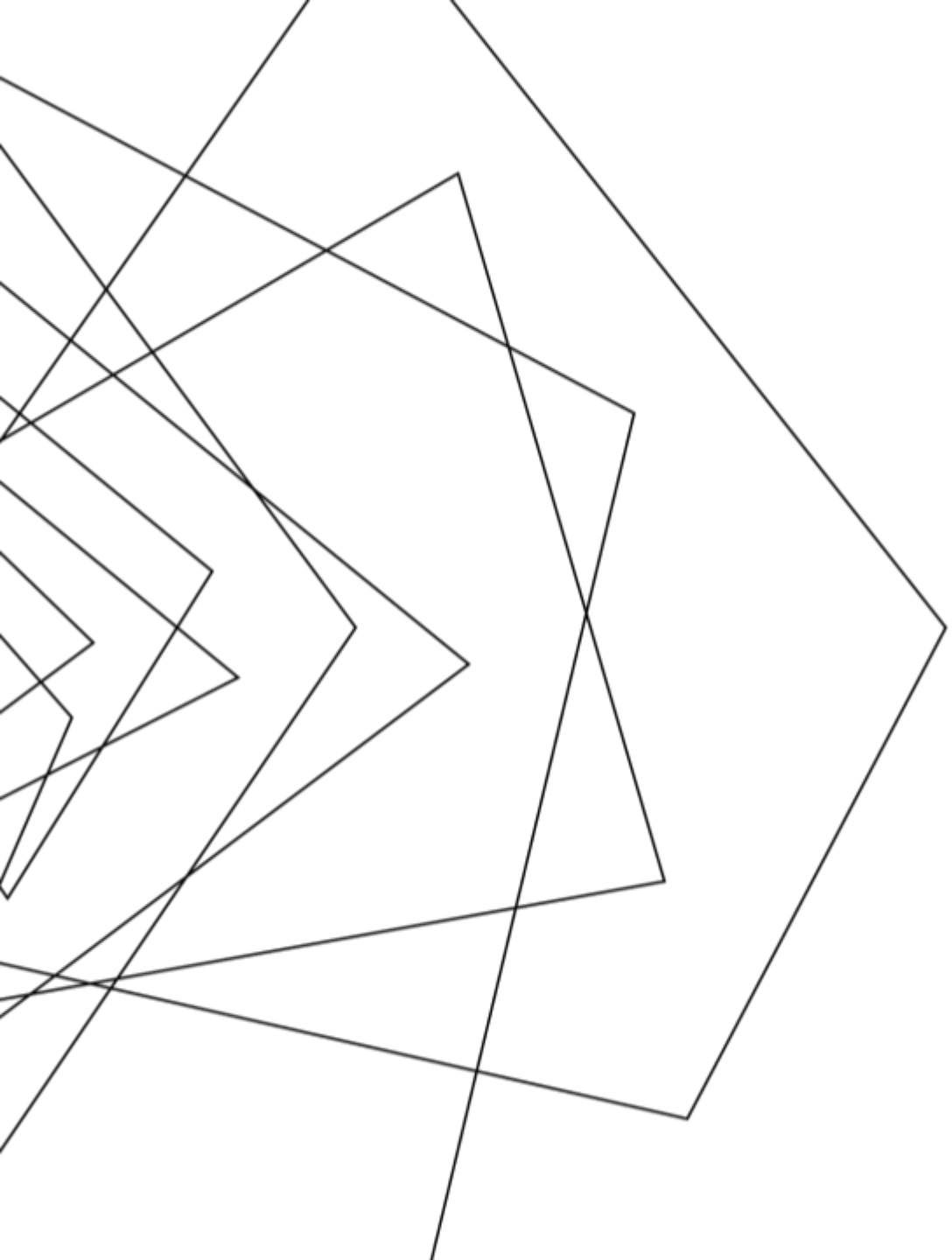
in OCaml

Practical exercises demonstrating type inference and type checking.

WEEK 5

Delving into advanced type theory topics such as dependent types and type constructors.

Exploring practical applications of dependent types in OCaml.



WEEK 6-7

Introduction to functional programming paradigms

Learning advanced techniques like continuation-passing style, monads, and concurrency

Writing concurrent programs in OCaml using different concurrency models.

METHODOLOGY

Reading Material will be sent at the beginning of every week.
There will be a variable number of discussion hours within that week.
At the end of the week, a programming assignment submission will be expected.

REFERENCE BOOKS

Real World OCaml: Functional programming for the masses by Yaron Minsky, Anil Madhavapeddy, and Jason Hickey

Types and Programming Languages by Benjamin C. Pierce

EVALUATION CRITERIA

•We will hold a mid-term and an end term test. The performance of the mentees in these tests will be the main criteria for ratification.

•The rough write-ups and assignments submitted by the mentees will also be the criteria

THANK YOU