10. Function Pointers

Hengfeng Wei (魏恒峰)
hfwei@nju.edu.cn

Dec. 05, 2024

Review

Pointer Arrays (`char *musicians[]`)

Pointers and 2D Arrays (`int (*scores)[]`)

Overview

Function Pointers

Why Function Pointers?

3.15

1 **object**

region of data storage in the execution environment, the contents of which can represent values

2 **Note 1 to entry:** When referenced, an object may be interpreted as having a particular type; see 6.3.2.1.

6.2.5 Types

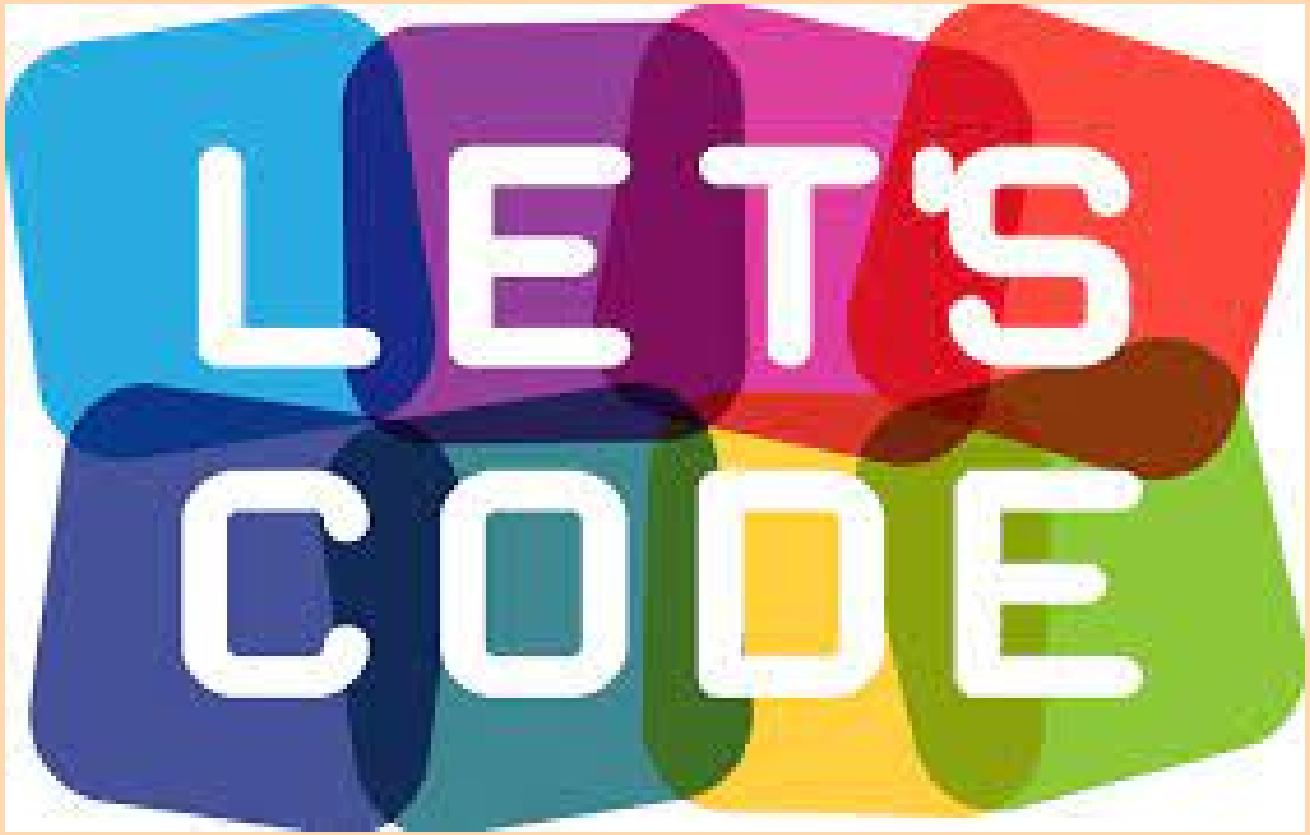
1 The meaning of a value stored in an object or returned by a function is determined by the *type* of the expression used to access it. (An identifier declared to be an object is the simplest such expression; the type is specified in the declaration of the identifier.) Types are partitioned into *object types* (types that describe objects) and *function types* (types that describe functions). At various points within a

Making Functions as Data (Objects)

Why they say...

Functions are first class citizens?

args **return vals** **assignment** **in array**



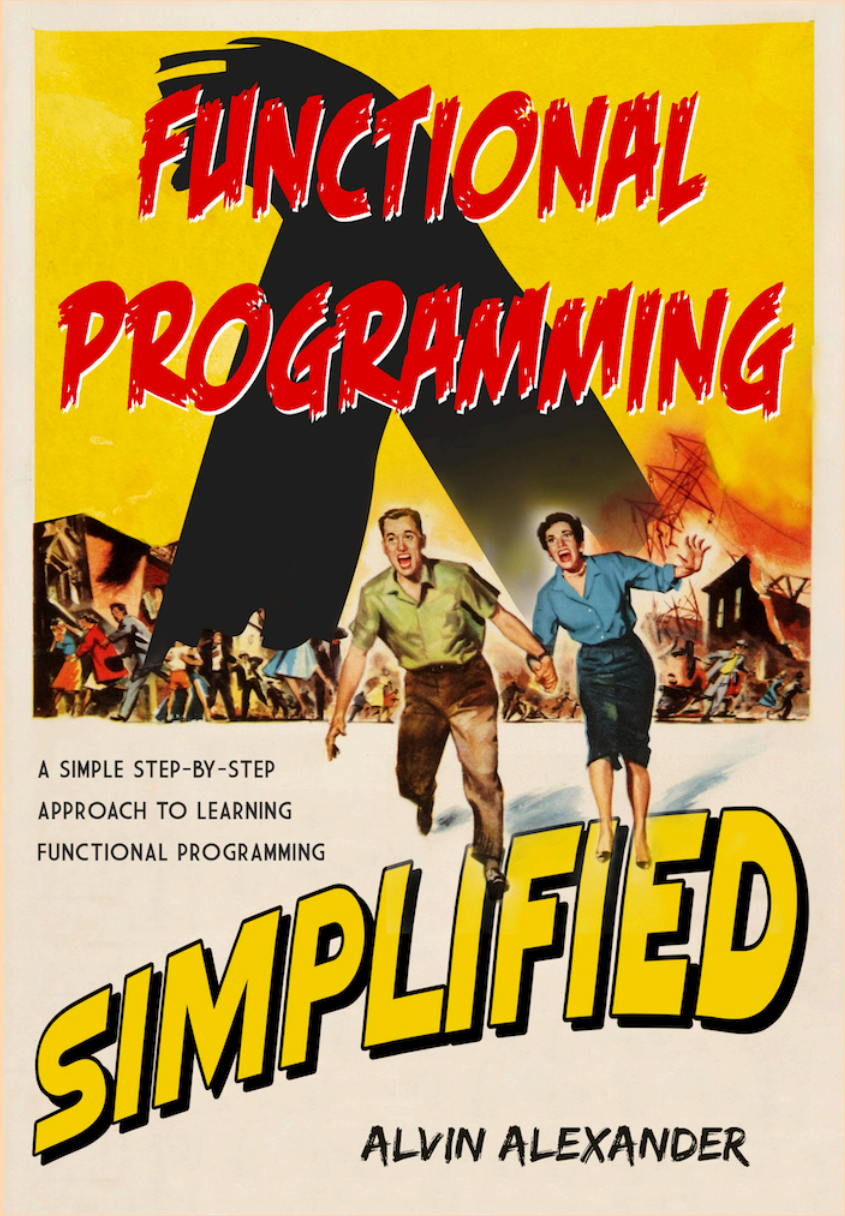
integrate.c **sort.c** **bsearch.c**

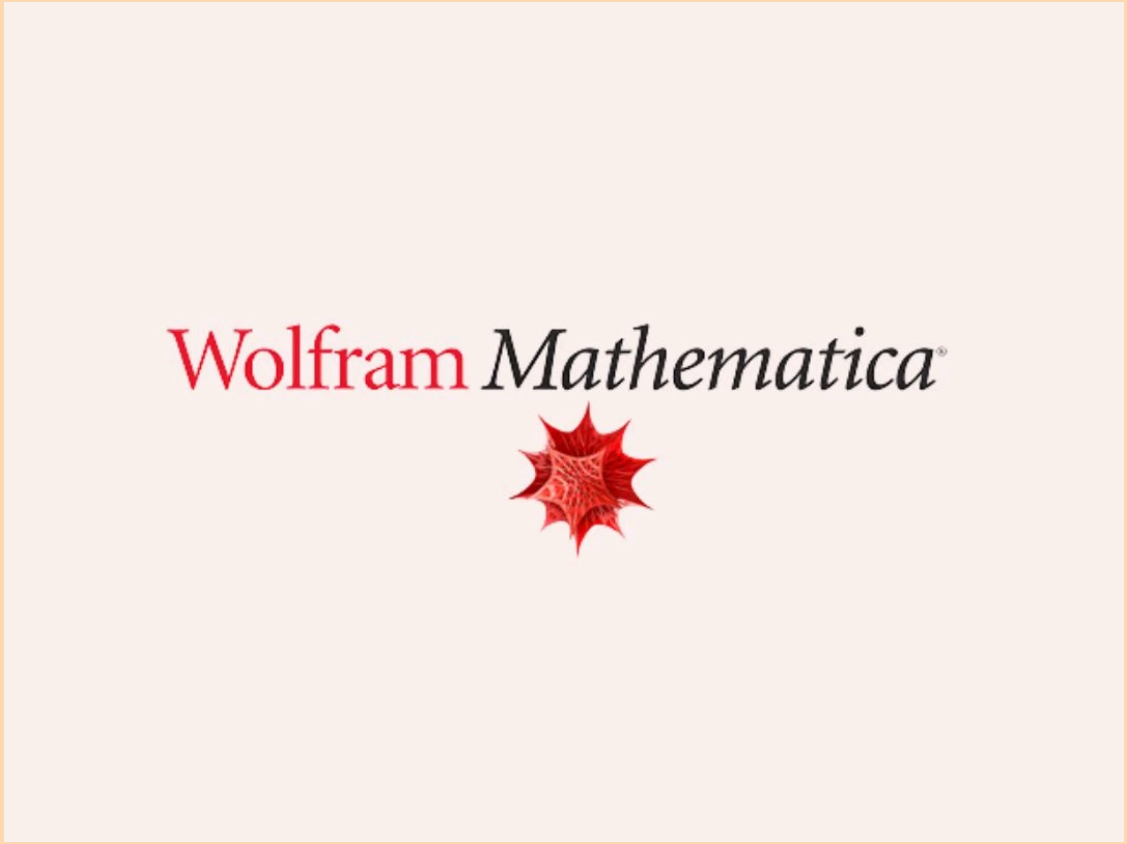
$$\int_a^b f(x)\mathrm{d}x \approx \sum_{i=0}^{n-1} f(x_i)(\frac{b-a}{n})$$

$$a = x_0 < x_1 < \cdots x_i < \cdots < x_{n-1} < x_n = b$$

$$x_i = a + \frac{b-a}{n} \cdot i$$

It is an Abstraction!!!


FUNCTIONAL PROGRAMMING
A SIMPLE STEP-BY-STEP APPROACH TO LEARNING FUNCTIONAL PROGRAMMING
SIMPLIFIED
ALVIN ALEXANDER


Wolfram Mathematica

decl.c


SEE
YOU
NEXT
WEEK