Farewell from Computer Science 211

Road map

Farewell from Computer Science 211

What's was it all about?

Road map

Farewell from Computer Science 211

What's was it all about?

Topics covered

Road map

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Road map

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Wanna see some Rust?

Up next

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Wanna see some Rust?

What CS 211 was all about (1/2)

From the course abstract:

- CS 211 teaches foundational software design skills at a small-to-medium scale. We will grow from writing single functions to writing interacting systems of several components.
- We aim to provide a bridge from the student-oriented *HtDP* languages (that is, CS 111) to real, industry-standard languages and tools. Like C11, the UNIX shell, Make, C++14, and CLion.
- In the first half...

What CS 211 is all about (2/2)

From the course abstract:

- In the first half, ~~you'll learn~~you learned the basics of imperative programming and manual memory management using the C programming language. This will help you form connections between the high-level programming concepts you learned in CS 111 and the low-level machine concepts you will learn in CS 213.
- In the second half, ~~we'll transition~~we transitioned to C++, which provides abstraction mechanisms such as classes and templates that we use to express our design ideas. ~~We'll learn~~We learned how to define our own, new types that act like the built-in ones.

What CS 211 is all about (2/2)

From the course abstract:

- In the first half, ~~you'll learn~~you learned the basics of imperative programming and manual memory management using the C programming language. This will help you form connections between the high-level programming concepts you learned in CS 111 and the low-level machine concepts you will learn in CS 213.
- In the second half, ~~we'll transition~~we transitioned to C++, which provides abstraction mechanisms such as classes and templates that we use to express our design ideas. ~~We'll learn~~We learned how to define our own, new types that act like the built-in ones.
- Topics included...

Up next

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Wanna see some Rust?

Topics

- Language mechanisms
  - New syntax for functional programming: expressions, values, conditionals, variables, functions
  - Imperative programming
    - Statements: sequencing, iteration
    - Mutation: objects, assignment
  - Memory allocation on the stack and the heap
  - Representing information with structs, arrays, pointers
  - Static types, type erasure, generics
- Design techniques
  - Data abstraction: defining our own types
  - Memory management via ownership and borrowing
  - RAII: Resource Acquisition Is Initialization
- Engineering practices
  - Testing: for gaining confidence in our software
  - Debugging: to see what's happening in memory
  - The Unix shell: a compositional user interface

Up next

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Wanna see some Rust?

Relative homework difficulties

HW	My Difficulty	Your Difficulty
1	3	?
2	5	?
3	7	?
4	11	?
5	6	?
6	8	?
FP	*	?

Up next

Farewell from Computer Science 211

What's was it all about?

Topics covered

How difficult was it?

Wanna see some Rust?

What is Rust?

- A low-level, systems programming language (like C++),

What is Rust?

- A low-level, systems programming language (like C++),
- but with high-level features (like algebraic datatypes),

What is Rust?

- A low-level, systems programming language (like C++),
- but with high-level features (like algebraic datatypes),
- and memory safety (no UB!).

What is Rust?

- A low-level, systems programming language (like C++),
- but with high-level features (like algebraic datatypes),
- and memory safety (no UB!).

Slogan: “Fearless concurrency”