0: Overview

Introduction to Computer Science:

A Mathematical Perspective

Prabhakar Ragde

University of Waterloo
Waterloo, Ontario, Canada

What This Course Is About

What This Course Is About

- (Apologies and thanks)

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...
- I first used Racket with my own children

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...
- I first used Racket with my own children
- I realized it could be used at the University of Waterloo

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...
- I first used Racket with my own children
- I realized it could be used at the University of Waterloo
- Every undergraduate student in the Faculty of Mathematics must take two computer science courses in first year

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...
- I first used Racket with my own children
- I realized it could be used at the University of Waterloo
- Every undergraduate student in the Faculty of Mathematics must take two computer science courses in first year
- What to do about diversity?

What This Course Is About

- (Apologies and thanks)
- This is a first course in **computer science**, not programming
- Traditionally a first course in CS is *only* programming
- We can do better because we are using the Racket language
- My history with math/CS and with Racket...
- I first used Racket with my own children
- I realized it could be used at the University of Waterloo
- Every undergraduate student in the Faculty of Mathematics must take two computer science courses in first year
- What to do about diversity?
- Racket (a **functional** programming language) was the solution

Racket

- https://www.racket-lang.org

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features
- Advanced Student: good enough for lots of CS

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features
- Advanced Student: good enough for lots of CS
- Contains powerful features that let us build new capabilities

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features
- Advanced Student: good enough for lots of CS
- Contains powerful features that let us build new capabilities
- Full Racket: allows creation of more languages

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features
- Advanced Student: good enough for lots of CS
- Contains powerful features that let us build new capabilities
- Full Racket: allows creation of more languages
- DrRacket: an excellent development environment

Racket

- `https://www.racket-lang.org`
- Racket is a family of programming languages
- The five Racket teaching languages
- Beginning Student: small, good enough to get started
- Next level: still small, introduces new features
- Advanced Student: good enough for lots of CS
- Contains powerful features that let us build new capabilities
- Full Racket: allows creation of more languages
- DrRacket: an excellent development environment
- Racket has changed my professional life...

Logistics of this course

Logistics of this course

- Based on a graduate course from UW's MMT program

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available

Logistics of this course

- Based on a graduate course from UW's MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available
- All programming questions available

Logistics of this course

- Based on a graduate course from UW's MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available
- All programming questions available
- Each question has associated evaluation program

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available
- All programming questions available
- Each question has associated evaluation program
- Final version of your solution emailed to tutors

Logistics of this course

- Based on a graduate course from UW’s MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available
- All programming questions available
- Each question has associated evaluation program
- Final version of your solution emailed to tutors
- Grade based on $\sim$20 questions, oral defense

Logistics of this course

- Based on a graduate course from UW's MMT program
- Emphasis on the connections between math and CS
- Eight lecture modules
- Written syllabus available
- All lecture slides available
- All programming questions available
- Each question has associated evaluation program
- Final version of your solution emailed to tutors
- Grade based on $\sim$20 questions, oral defense
- Some model solutions may be made available