COMP 302 Winter 2020 Final Review

Prakash Panangaden

School of Computer Science
McGill University

McGill University,Montréal, January 2020

Goodbye to COMP 302

- It is very sad to be ending the class in this way.

Goodbye to COMP 302

- It is very sad to be ending the class in this way.
- But there are much worse things happening these days.

Goodbye to COMP 302

- It is very sad to be ending the class in this way.
- But there are much worse things happening these days.
- Today I will give a capsule summary of the course.

Goodbye to COMP 302

- It is very sad to be ending the class in this way.
- But there are much worse things happening these days.
- Today I will give a capsule summary of the course.
- Then I will describe the final exam.

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.
2. I said that I would **never** use slides again.

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.
2. I said that I would **never** use slides again.
3. I lied.

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.
2. I said that I would **never** use slides again.
3. I lied.
4. There is a MyCourses page with links to the course web site and grades.

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.
2. I said that I would **never** use slides again.
3. I lied.
4. There is a MyCourses page with links to the course web site and grades.
5. The **final exam is submitted through myCourses.**

Remember the course administration details

1. `cs.mcgill.ca/~prakash/Courses/302/comp302.html`
   Lecture notes, assignments and solutions will be posted there.
2. I said that I would **never** use slides again.
3. I lied.
4. There is a MyCourses page with links to the course web site and grades.
5. The **final exam is submitted through myCourses.**
6. Instructors and students will communicate using Piazza.

Grading

- 8 assignments : 24% Submitted through an automated system using LearnOCaml.

Grading

- 8 assignments : 24% Submitted through an automated system using LearnOCaml.
- 3 quizzes : 6% using the myCourses system.

Grading

- 8 assignments : 24% Submitted through an automated system using LearnOCaml.
- 3 quizzes : 6% using the myCourses system.
- 1 in-class midterm: 10% of your grade,

Grading

- 8 assignments : 24% Submitted through an automated system using LearnOCaml.
- 3 quizzes : 6% using the myCourses system.
- 1 in-class midterm: 10% of your grade,
- Final exam: 60% of your total grade.

What did we cover?

1. Recursion

What did we cover?

1. Recursion
2. How to think about it

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking
10. Polymorphic typing

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking
10. Polymorphic typing
11. Parsing and compilers

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking
10. Polymorphic typing
11. Parsing and compilers
12. Streams and lazy evaluation

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking
10. Polymorphic typing
11. Parsing and compilers
12. Streams and lazy evaluation
13. Subtyping

What did we cover?

1. Recursion
2. How to think about it
3. Inductively defined types: lists, trees
4. Environments and binding
5. Higher-order functions
6. Operational semantics
7. Updatable data: references
8. Closures and objects
9. Typing rules and type checking
10. Polymorphic typing
11. Parsing and compilers
12. Streams and lazy evaluation
13. Subtyping
14. Subtyping and inheritance in Java

Types

1. Classify values

Types

1. Classify values
2. and expressions

Types

1. Classify values
2. and expressions
3. and functions and procedures

Types

1. Classify values
2. and expressions
3. and functions and procedures
4. in order to *restrict* what can be expressed.

Types

1. Classify values
2. and expressions
3. and functions and procedures
4. in order to *restrict* what can be expressed.
5. Give up expressive power for

Types

1. Classify values
2. and expressions
3. and functions and procedures
4. in order to *restrict* what can be expressed.
5. Give up expressive power for
6. guarantees of good behaviour.

What to understand

1. Names: binding and scoping

What to understand

1. Names: binding and scoping
2. Evaluation rules: expressions $\rightarrow$ values

What to understand

1. Names: binding and scoping
2. Evaluation rules: expressions $\rightarrow$ values
3. Typing rules,

What to understand

1. Names: binding and scoping
2. Evaluation rules: expressions $\rightarrow$ values
3. Typing rules,
4. which may not be exclusive.

Where do we go from here?

1. Intimate connection between logic and computation: the Curry-Howard isomorphism

Where do we go from here?

1. Intimate connection between logic and computation: the Curry-Howard isomorphism
2. New directions in type theory: guaranteeing security

Where do we go from here?

1. Intimate connection between logic and computation: the Curry-Howard isomorphism
2. New directions in type theory: guaranteeing security
3. New logics and new programming paradigms: linear logic

Where do we go from here?

1. Intimate connection between logic and computation: the Curry-Howard isomorphism
2. New directions in type theory: guaranteeing security
3. New logics and new programming paradigms: linear logic
4. Probabilistic programming languages designed for machine learning

Overview of OCaml

1. Functional - functions are the main entities.
2. Higher-order - functions may take other functions as arguments and
3. may even return functions as results.
4. Typed - every entity has a type.
5. Types are described in their own little language; types are not just the basic types
6. but are polymorphic and one can have new user-defined types.

Structure of the exam

- There are 6 questions in total.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.
- One is straightforward and one is not.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.
- One is straightforward and one is not.
- One question on higher-order functions.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.
- One is straightforward and one is not.
- One question on higher-order functions.
- One question on higher-order functions with refs.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.
- One is straightforward and one is not.
- One question on higher-order functions.
- One question on higher-order functions with refs.
- One question where you have to give an **informal** derivation of the polymorphic type of a function.

Structure of the exam

- There are 6 questions in total.
- 2 questions on recursion on lists.
- One is straightforward and one is not.
- One question on higher-order functions.
- One question on higher-order functions with refs.
- One question where you have to give an **informal** derivation of the polymorphic type of a function.
- One collection of true/false questions. No explanations are needed.

Topics

1. Recursion

Topics

1. Recursion
2. Inductively defined types: lists, trees

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects
7. Typing rules and type checking

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects
7. Typing rules and type checking
8. Polymorphic typing

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects
7. Typing rules and type checking
8. Polymorphic typing
9. Streams and lazy evaluation

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects
7. Typing rules and type checking
8. Polymorphic typing
9. Streams and lazy evaluation
10. Subtyping

Topics

1. Recursion
2. Inductively defined types: lists, trees
3. Environments and binding
4. Higher-order functions
5. Updatable data: references
6. Closures and objects
7. Typing rules and type checking
8. Polymorphic typing
9. Streams and lazy evaluation
10. Subtyping

What isn't here isn't on the exam.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.
- The solutions must be written in **plain text** in **one single file.**

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.
- The solutions must be written in **plain text** in **one single file.**
- I **DO NOT WANT** pdf, latex, Word, handwritten scans or anything else. I don't want each question in a new file.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.
- The solutions must be written in **plain text** in **one single file.**
- I **DO NOT WANT** pdf, latex, Word, handwritten scans or anything else. I don't want each question in a new file.
- I will NOT accept an exam emailed to me or the TAs.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.
- The solutions must be written in **plain text** in **one single file.**
- I **DO NOT WANT** pdf, latex, Word, handwritten scans or anything else. I don't want each question in a new file.
- I will NOT accept an exam emailed to me or the TAs.
- Don't wait for the last minute to upload and then tell me that you had an internet connection problem. I will not accept an emailed exam.

How to submit

- At 6:30pm on the 22nd of April the exam will be posted on the course web page. There will be links in Piazza and on myCourses to this exam.
- The exam is a pdf file. Make sure you have a pdf viewer.
- The solutions must be written in **plain text** in **one single file**.
- I **DO NOT WANT** pdf, latex, Word, handwritten scans or anything else. I don't want each question in a new file.
- I will NOT accept an exam emailed to me or the TAs.
- Don't wait for the last minute to upload and then tell me that you had an internet connection problem. I will not accept an emailed exam.
- Once you upload your solution the system will not accept updates.

Submission

- I will create a new dummy **assignment** with zero weight where you will upload the final exam solutions. myCourses only allows students to submit assignments, so we have to pretend that the final is an assignment.

Submission

- I will create a new dummy **assignment** with zero weight where you will upload the final exam solutions. myCourses only allows students to submit assignments, so we have to pretend that the final is an assignment.
- The actual final exam marks will be entered in the proper place in the Grade book.