INTRODUCTION

PRINCIPLES OF PROGRAMMING LANGUAGES

Norbert Zeh
Winter 2019
Dalhousie University
GOAL OF THIS COURSE

Encourage you to become better programmers (≠ Make you better programmers)
Encourage you to become better programmers
Encourage you to become better programmers

(≠ Make you better programmers)
Programming paradigms
• Imperative vs functional vs logic programming

Programming language abstractions
• Variable binding, parameter passing, lifetime of variables, …

What drives these design decisions
• Call stacks, closures, thunks, memory management, garbage collection, …

Compilation, interpretation, and formal languages
• …
Programming paradigms

- Imperative vs functional vs logic programming
Programming paradigms
  • Imperative vs functional vs logic programming

Programming language abstractions
  • Variable binding, parameter passing, lifetime of variables, ...
Programming paradigms

- Imperative vs functional vs logic programming

Programming language abstractions

- Variable binding, parameter passing, lifetime of variables, ...

What drives these design decisions

- Call stacks, closures, thunks, memory management, garbage collection, ...
Programming paradigms

- Imperative vs functional vs logic programming

Programming language abstractions

- Variable binding, parameter passing, lifetime of variables, ...

What drives these design decisions

- Call stacks, closures, thunks, memory management, garbage collection, ...

Compilation, interpretation, and formal languages
THIS COURSE IN A NUTSHELL

Programming paradigms (1)

• Imperative vs functional vs logic programming

Programming language abstractions (3)

• Variable binding, parameter passing, lifetime of variables, ...

What drives these design decisions (3)

• Call stacks, closures, thunks, memory management, garbage collection, ...

Compilation, interpretation, and formal languages (2)
COURSE MATERIAL

Books:

- Hopcroft et al. *Introduction to Automata Theory.* (optional)
- Tucker and Noonan. *Programming Languages.* (optional)
- More relevant books on course website. Some available online.

Slides:

- On website

Website:

- http://www.cs.dal.ca/~nzeh/Teaching/3136

Email:

- nzeh@cs.dal.ca
Class:
  • Mon, Wed, Fri: 3:30–4:30

Office hours:
  • Mon, Wed, Fri 12:00–2:00
  • Mona Campbell 4246

TAs:
  • TBA
EVALUATION

(A)ssignments:

- 8 assignments
- 4 programming assignments
- 4 theory assignments
- Each has equal weight
- 3 best programming assignments count, 3 best theory assignments count

(M)idterm

(F)inal

Grade = \max (40\% A + 20\% M + 40\% F; 40\% A + 60\% F)$
EVALUATION

(A)ssignments:
  • 8 assignments
  • Each has equal weight

(M)idterm

(F)inal
EVALUATION

(A)ssignments:

• 8 assignments
  • 4 programming assignments
  • 4 theory assignments

• Each has equal weight

(M)idterm

(F)inal
(A)ssignments:

- 8 assignments
  - 4 programming assignments
  - 4 theory assignments
- Each has equal weight
- 3 best programming assignments count, 3 best theory assignments count

(M)idterm

(F)inal
EVALUATION

(A)ssignments:

• 8 assignments
  • 4 programming assignments
  • 4 theory assignments

• Each has equal weight

• 3 best programming assignments count, 3 best theory assignments count

(M)idterm

(F)inal

Grade = max(40% \cdot A + 20% \cdot M + 40% \cdot F, 40% \cdot A + 60% \cdot F)
Implement a Scheme interpreter in Scheme

- Implementation language: Scheme R6RS
- Language to implement: A subset of Scheme
THE GREAT INTERPRETER PROJECT OF 2019 (3 PROGRAMMING ASSIGNMENTS)

Lexical analysis
- Compile regular expression to NFA
- Translate NFA to DFA & minimize it
- Build a greedy scanner

Syntactic analysis
- Build a recursive descent parser

Semantic analysis
- Augment parser with semantic analysis & translate to bytecode

Execution of bytecode
- Implement simple bytecode interpreter
GROUP WORK ON ASSIGNMENTS

- Work in groups of up to 3 students (strongly encouraged!)

- Each group submits one joint assignment. Every group member gets the same marks.

- Group composition may change between assignments.

- No exchange of information between groups!
Late submissions:

- ... are not accepted.
- Exceptions: You were sick or agreed on an extension with me beforehand (e.g., if there’s a wedding in the family)

Academic honesty:

- https://www.dal.ca/dept/university_secretariat/academic-integrity.html
- No exchange of information between groups on assignments.
- All reference material (book, web, ...) must be acknowledged.
- No need to reference material presented in class.
- Any suspected case of plagiarism is referred to the Academic Integrity Officer.
We believe that inclusiveness is fundamental to education and learning.

Every person has the right to be respected and safe.

Mysogyny and disrespectful behaviour on campus, in the wider community or on social media is not acceptable.

We stand for equality and hold ourselves to a higher standard.

Take an active role:

- Be ready: Don’t remain silent.
- Identify the behaviour, avoid labelling, name-calling or blame.
- Appeal to principles, particularly with friends and co-workers.
- Set limits.
- Find an ally and be an ally, lead by example.
- Be vigilant.
LANGUAGES YOU’LL NEED

Used extensively:
  • Scheme
  • Prolog

Mentioned:
  • C/C++
  • Python
  • Java
  • Scala
  • Ruby
  • Haskell
  • …