Programming Languages CSCI-4430 Fall 2020

www.cs.rpi.edu/~milanova/csci4430/

Ana Milanova
https://rensselaer.webex.com/meet/milana2, milanova@cs.rpi.edu
Konstantin Kuzmin
https://rensselaer.webex.com/meet/kuzmik2, kuzmik2@rpi.edu

1

Lecture Outline

- Introduction: the rules!
  - Strangest Proglang ever
- Programming language spectrum
- Why study programming languages?
- Compilation and interpretation

Read: Scott Chapter 1

2

Introduction

- Course webpage
  https://www.cs.rpi.edu/~milanova/csci4430
  - Schedule, Notes, Reading
    - Schedule, lecture slides, assigned reading, and homework links
  - Submitty
    - Homework submission and grades (Rainbow grades)
    - Discussion forum! Announcements!
    - Please, submit your TIME ZONE!

3

Introduction

- Required textbook
  - **Programming Language Pragmatics,** 4th Edition, by Michael Scott, Morgan Kaufmann, 2015
- Recommended textbook
  - **Compilers: Principles, Techniques, and Tools,** 2nd Edition, by A. Aho, M. Lam, R. Sethi and J. Ullman, Addison Wesley, 2007 (as known as "The Dragon Book")

4

Introduction

- Syllabus
  https://www.cs.rpi.edu/~milanova/csci4430/syllabus.html
  Topics, outcomes, policies, and grading
- 2 midterm exams and a final exam: 47%
- 7 homework assignments: 42%
- 8 quizzes: 8%
- 3% office hours check-in

5

Introduction

- Lectures will be WebEx Events on Tuesdays and Fridays 2:30pm – 4:20pm Eastern Time
  - You will receive invitation for each lecture
  - Begin with Q/A on prior topics/homework, or a quiz
  - Next, pre-recorded lecture and some Q/A
- Lectures will be available shortly before scheduled lecture:
  - Recording: https://mediasite.mms.rpi.edu/Mediasite5/Channel/programming_languages
  - PDF notes: https://www.cs.rpi.edu/~milanova/csci4430/schedule.html

6

Introduction

- Homework is due at 2pm on the due date
- Submit typed homework as a PDF electronically in Submitty
- Submit programming homework in Submitty for autograding
- Homework, including submission instructions, will be posted at
  - https://www.cs.rpi.edu/~milanova/csci4430/schedule.html

7

Introduction

- Homework is due at 2pm on the due date
- 6 late days in total
- 2 late days at most per homework
- Extensions only with a formal excuse note from your class dean. See syllabus for details.

8

Introduction

- Quizzes
  - 8 (or so) quizzes during regularly scheduled class hours
  - Will cover material of previous weeks
  - Work in groups (up to 6 people) is encouraged
    - Do not post on sites/channels globally visible to class
  - If you are unable to "attend" class throughout the term, email us to schedule an alternative time for quizzes and exams. (Syllabus describes procedure.)

9

Introduction

- Quiz or exam makeup will be arranged only after we have received an excuse note from your class dean. See syllabus for details.

10

Introduction

- Office hours
  - We plan for ample office hours on Mondays, Wednesdays and Thursdays
  - Instructor office hours right after class
  - We'll require weekly office hour "check-ins" starting at week 3 for at least 10 weeks
  - TA and mentor office hours via Submitty queues

11

Other Notes

- Asking questions
  - First, go to Submitty forum
    - Do not post code on forum
    - You cannot post code to any website
  - Second, go to office hours
    - Sessions are individual, run through Submitty queues, so you can
  - Third (last resort): proglang-help@cs.lists.rpi.edu goes to instructors
- We will not be answering questions coming in late at night or in the morning on day HW is due

12

Other Notes

- Submitty forum
  - Announcements – check regularly
  - Ask all non-personal questions on the forum
  - Check out prior messages before you post a question – the answer is probably already there
- Mailing list proglang@cs.lists.rpi.edu (instructors)
  - Personal questions (extensions, grade disputes, etc.)
  - Unsolicited debugging emails to instructors or mailing list will likely go unanswered!

Other Notes

- Debugging and homework help in office hours
  - Instructor, TA and mentor office hours will be finalized by beginning of next week

Academic Integrity

- In short, do not copy and do not post your solutions/code on public forums or repos
- Excessive similarities between homework submissions will be considered cheating and handled accordingly
- I trust you. Submitty has advanced plagiarism detection tools that course stuff runs regularly

How to Study

- Read textbook chapter in advance of lecture
  - Chapters are announced on Schedule
- Read/listen lecture and read textbook chapter immediately after class
  - Lecture pdfs and recording will be available shortly before scheduled class hours
- Solve exercises in lectures
- Form study groups
- ASK QUESTIONS – in class, on forum

How to Study

- A 15 min presentation from the ALAC:
- https://mediasite.mms.rpi.edu/Mediasite5/Play/3c69d5096dc5494eadcaba2b9c99189f1d

Course Topics

- Programming language syntax: Scanning and parsing
- Programming language semantics: Attribute grammars
- Naming, binding and scoping
- Data abstraction and types
- Control abstraction and parameter passing
- Concurrency
- Logic-oriented language: Prolog
- Functional languages: Scheme and Haskell
- Imperative languages
  - An object-oriented language: Java
  - A dynamic language: Python

Course Topics

- Schedule at www.cs.rpi.edu/~milanova/csci4430/schedule.html
- Lists major and minor topics
- Homework links, dates and due dates
- Quiz and exam schedule

19

Lecture Outline

- Introduction to the course
- Programming language spectrum
- Why study programming languages?
- Compilation and interpretation

20

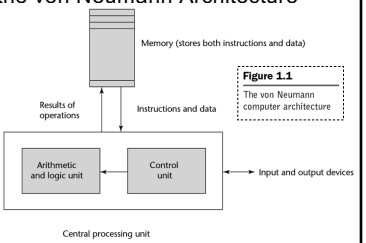
The Programming Language Spectrum

- Imperative languages
  - Von Neumann languages: Fortran, C,…
  - Object-oriented languages: Java, C++, Smalltalk,…
  - Dynamic languages: Perl, Python, PHP,…
- Declarative languages
  - Functional languages: Scheme/Lisp, ML, Haskell
  - Logic languages: Prolog
  - There are other declarative languages: e.g., dataflow languages

21

The Programming Language Spectrum

- Imperative languages
  - Evolved from the von Neumann Architecture
  - Variable
  - Assignment Statement



22

The Programming Language Spectrum

- Imperative languages
  - Most widely popular programming style
    - FORTRAN, C, C++, C#, Java, Python, Visual BASIC, Perl, JavaScript, Ruby, etc.
  - Variable and assignment statement are central concepts
  - Program is a sequence of statements:

```
j := i – j;
k := j * l;
```

    - Execution is a sequence of transitions on memory state

23

The Programming Language Spectrum

- FORTRAN was invented in mid-1950
- John Backus, the inventor of FORTRAN, wrote the following paper in 1979: "Can programming be liberated from the von Neumann style? A functional style and its algebra of programs"
  - Problems with imperative languages
    - Difficult to understand programs
    - Difficult to reason about correctness of programs

24

The Programming Language Spectrum

- Functional Programming
  - Main alternative to imperative programming
    - Lisp/Scheme, ML/OCaml, Haskell
  - Program consists of function definitions + evaluation expr

```
(fun3 (fun2 (fun1 data)))
(fun3 (fun2 data2))
(fun3 data3)
data4
```

    - Execution is a sequence of function applications (i.e., reductions)
- Logic Programming
  - Perform queries against knowledge base
  - Prolog, Datalog, SQL

25

An Example: Inner Product

- Inner product in FORTRAN:

```
1. C := 0;
2. for I := 1 step 1 until N do
3.   C := C + a[I]*b[I];
```

- Illustrates state-transition semantics

26

An Example: Inner Product

- Inner product in FP:

Function composition

Def IP = (Insert +) ° (ApplyToAll *) ° Transpose

IP <<1,2,3>,<6,5,4>> is
(Insert +) ((ApplyToAll *) (Transpose <<1,2,3>,<6,5,4>>))
(Insert +) ((ApplyToAll *) <<1,6>,<2,5>,<3,4>>)
(Insert +) <6,10,12>
28

- Illustrates reduction (applicative) semantics

27

Why Study Programming Languages

- Goal of the course: learn to analyze programming languages
  - What are the questions we ask when facing a new programming language
  - Helps learn new languages, choose the right language for a problem, understand language features, design better languages!

28

Lecture Outline

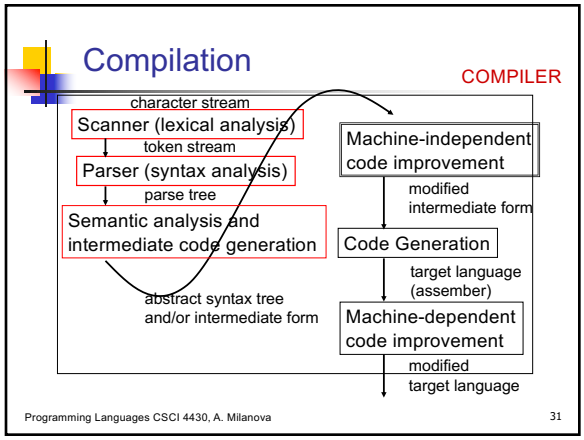
- Introduction to the course
- The programming language spectrum
- Why study programming languages
- Compilation and interpretation

29

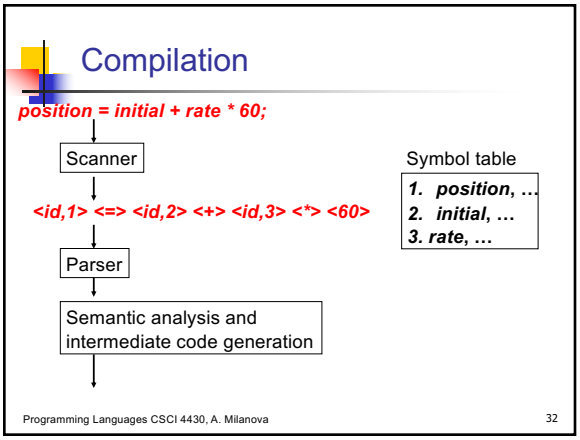
Compilation and Interpretation

- Compilation
  - Compiler
  - A "high-level" program is translated into executable machine code
- Interpretation
  - Interpreter
  - A program is translated and executed one statement at a time
- Hybrid interpretation
  - Both a compiler and an interpreter
  - A program is "compiled" into intermediate code; intermediate code is "interpreted"

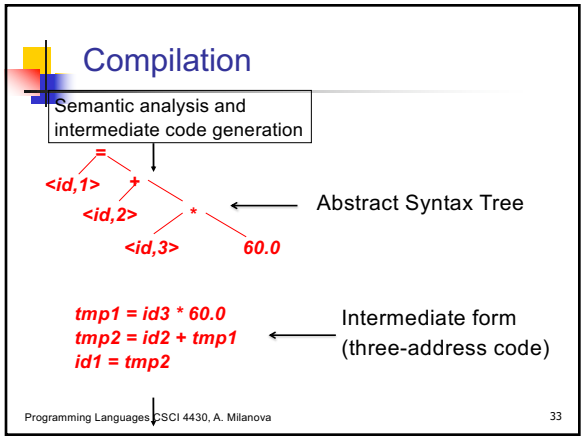
30



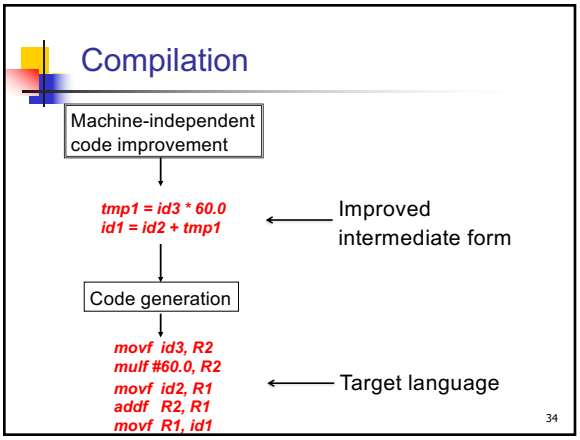
31



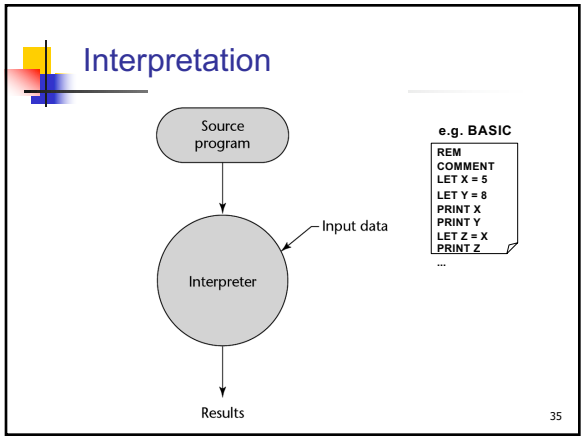
32



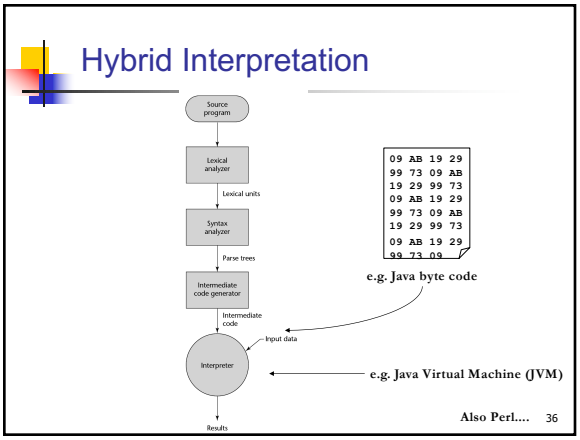
33



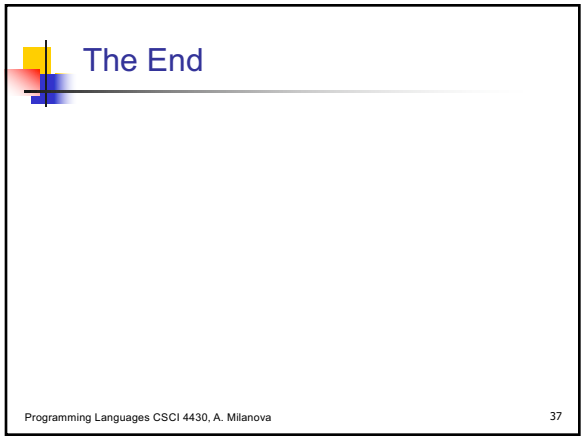
34



35



36



37