Course Mechanics

Introduction to Compiler Construction

Outline

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Outline

Website

Goal

Prerequisites

Staff

Lectures

Text

Grading Scheme

Software

Policies

Immediate Action Items

Topics Covered

Website

Website

`https://swamiiyer.net/cs451`

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)
- Course Info

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)
- Course Info
- Calendar

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)
- Course Info
- Calendar
- Lecture Material

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)
- Course Info
- Calendar
- Lecture Material
- Assignments

Website

`https://swamiiyer.net/cs451`

Stuff on the site:

- Announcements (landing page)
- Course Info
- Calendar
- Lecture Material
- Assignments
- Resources

Goal

Goal

Theory:

Goal

Theory:

- Scan a program into a stream of tokens

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit
- Analyze various programming constructs and generate code for them

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit
- Analyze various programming constructs and generate code for them
- Allocate physical registers to a program expressed in terms of virtual registers

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit
- Analyze various programming constructs and generate code for them
- Allocate physical registers to a program expressed in terms of virtual registers

Practice:

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit
- Analyze various programming constructs and generate code for them
- Allocate physical registers to a program expressed in terms of virtual registers

Practice:

- Develop a compiler called *j*-- targeting the stack-based Java Virtual Machine (JVM)

Goal

Theory:

- Scan a program into a stream of tokens
- Parse a program making its syntactic structure explicit
- Analyze various programming constructs and generate code for them
- Allocate physical registers to a program expressed in terms of virtual registers

Practice:

- Develop a compiler called *j*-- targeting the stack-based Java Virtual Machine (JVM)
- Develop a compiler called *iota* targeting the register-based Marvin machine

Prerequisites

Prerequisites

CS310 (Advanced Data Structures and Algorithms) *and*

Prerequisites

CS310 (Advanced Data Structures and Algorithms) *and*

CS420 (Intro. to the Theory of Computation) *or* CS622 (Theory of Formal Languages) *or*

Prerequisites

CS310 (Advanced Data Structures and Algorithms) *and*

CS420 (Intro. to the Theory of Computation) *or* CS622 (Theory of Formal Languages) *or*

Permission of the instructor

Staff

Staff

Instructor:

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)
- Website: `https://swamiiyer.net`

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)
- Website: `https://swamiiyer.net`
- Office Hours:

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)
- Website: `https://swamiiyer.net`
- Office Hours:
  - `Tue Thu 8:00 AM - 9:00 AM` and `2:30 PM - 3:30 PM` (in-person)

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)
- Website: `https://swamiiyer.net`
- Office Hours:
  - `Tue Thu 8:00 AM - 9:00 AM` and `2:30 PM - 3:30 PM` (in-person)
  - `Wed 10:00 AM - 12:00 PM` (remote)

Staff

Instructor:

- Name: Swami Iyer (Senior Lecturer I, Computer Science Department)
- Office: `M-3-201-14`
- Email: `siyer@cs.umb.edu` (start subject line with `[CS451]`)
- Website: `https://swamiiyer.net`
- Office Hours:
  - `Tue Thu 8:00 AM - 9:00 AM` and `2:30 PM - 3:30 PM` (in-person)
  - `Wed 10:00 AM - 12:00 PM` (remote)

Teaching Assistant (TA): Vijay Tummalapalli

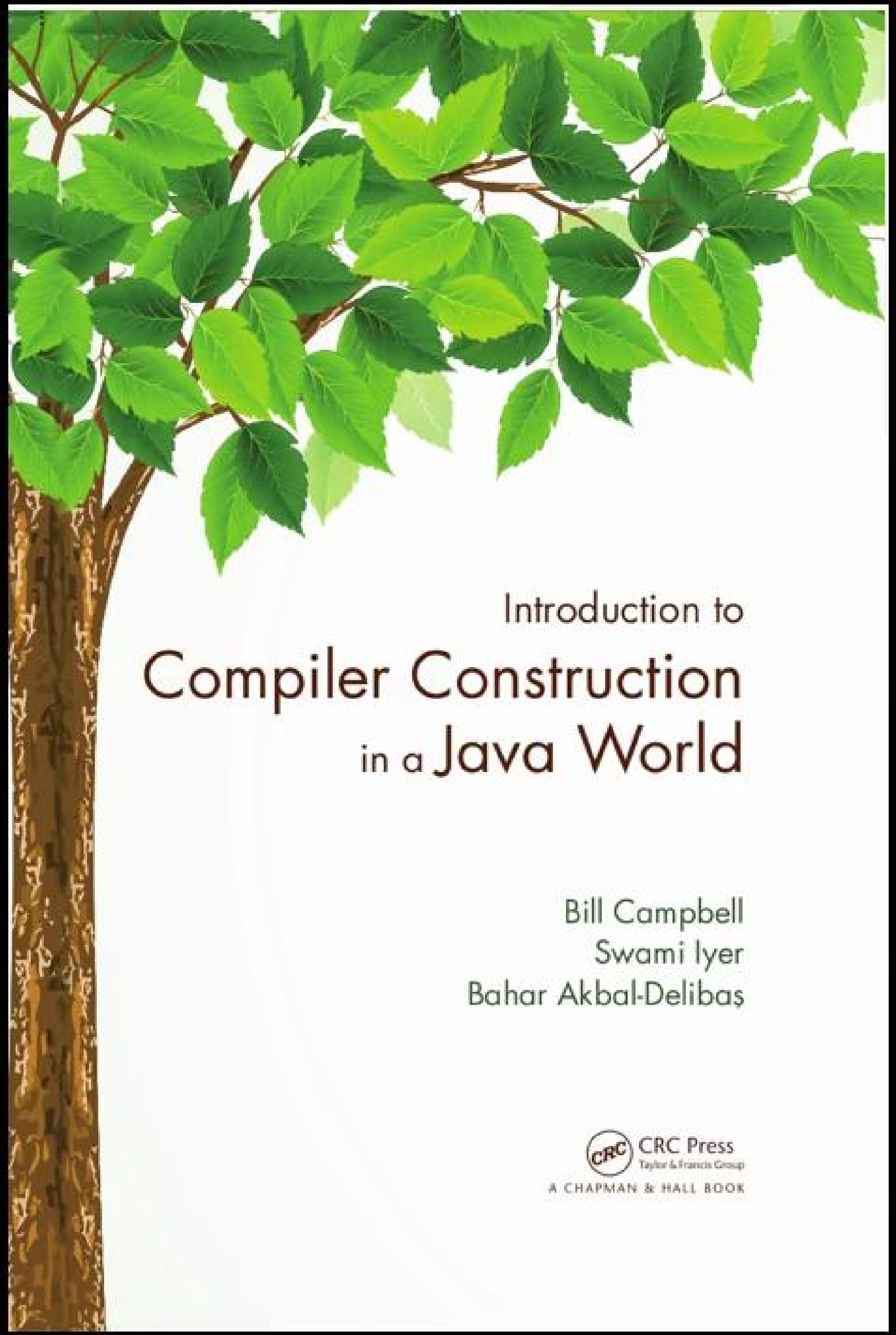
Lectures

Lectures

When	Where
Tue Thu 4:00 PM - 5:15 PM	Y-2-2110

Text

Text



Introduction to

Compiler Construction

in a Java World

Bill Campbell

Swami Iyer

Bahar Akbal-Delibaş

CRC Press

Taylor & Francis Group

A CHAPMAN & HALL BOOK

Grading Scheme

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j*-- and *iota* compilers

Grading Scheme

Assessment	**% of Final Grade**
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j*-- and *iota* compilers

Exams: Oct 30 and Dec 11, on theoretical aspects of a compiler

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j--* and *iota* compilers

Exams: Oct 30 and Dec 11, on theoretical aspects of a compiler

Participation: best 10 in-class quizzes

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j--* and *iota* compilers

Exams: Oct 30 and Dec 11, on theoretical aspects of a compiler

Participation: best 10 in-class quizzes

If each exam score is at least 87 (B+), the higher score will be taken as the exam average

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j*-- and *iota* compilers

Exams: Oct 30 and Dec 11, on theoretical aspects of a compiler

Participation: best 10 in-class quizzes

If each exam score is at least 87 (B+), the higher score will be taken as the exam average

Up to 1% extra points for completing the end-of-semester course evaluation

Grading Scheme

Assessment	% of Final Grade
Programming Assignments (1, 2, 3, 5, and best of 4 and 6)	30
Exams (2)	60
Participation	10

Programming assignments: biweekly, on enhancing the functionality of the *j*-- and *iota* compilers

Exams: Oct 30 and Dec 11, on theoretical aspects of a compiler

Participation: best 10 in-class quizzes

If each exam score is at least 87 (B+), the higher score will be taken as the exam average

Up to 1% extra points for completing the end-of-semester course evaluation

If overall score is within 0.5% of a higher grade, it will be elevated to that grade

Software

Software

Piazza

Software

Piazza

Gradescope

Software

Piazza

Gradescope

Programming Environment

Software

Piazza

Gradescope

Programming Environment

Zoom

Policies

Policies

Classroom

Policies

Classroom

Piazza

Policies

Classroom

Piazza

Makeup Exam

Policies

Classroom

Piazza

Makeup Exam

Assignment Deadline

Policies

Classroom

Piazza

Makeup Exam

Assignment Deadline

Regrade Request

Policies

Classroom

Piazza

Makeup Exam

Assignment Deadline

Regrade Request

Collaboration

Policies

Classroom

Piazza

Makeup Exam

Assignment Deadline

Regrade Request

Collaboration

Accommodations for Students with Disabilities

Policies

Classroom

Piazza

Makeup Exam

Assignment Deadline

Regrade Request

Collaboration

Accommodations for Students with Disabilities

Campus Closure

Immediate Action Items

Immediate Action Items

Sign up for Piazza

Immediate Action Items

Sign up for Piazza

Sign up for Gradescope

Immediate Action Items

Sign up for Piazza

Sign up for Gradescope

Set up the programming environment

Immediate Action Items

Sign up for Piazza

Sign up for Gradescope

Set up the programming environment

Fill out the course agreement available on Gradescope

Immediate Action Items

Sign up for Piazza

Sign up for Gradescope

Set up the programming environment

Fill out the course agreement available on Gradescope

Sign up for CS account

Topics Covered

Topics Covered

Chapter 1 (Compilation):

Topics Covered

Chapter 1 (Compilation):

- Preliminaries

Topics Covered

Chapter 1 (Compilation):

- Preliminaries
- Overview of the *j*-- to JVM compiler

Topics Covered

Topics Covered

Chapter 2 (Scanning):

Topics Covered

Chapter 2 (Scanning):

- Preliminaries

Topics Covered

Chapter 2 (Scanning):

- Preliminaries
- Handcrafting a Scanner

Topics Covered

Chapter 2 (Scanning):

- Preliminaries
- Handcrafting a Scanner
- Generating a Scanner

Topics Covered

Chapter 2 (Scanning):

- Preliminaries
- Handcrafting a Scanner
- Generating a Scanner
- JavaCC Scanner for *j*--

Topics Covered

Topics Covered

Chapter 3 (Parsing):

Topics Covered

Chapter 3 (Parsing):

- Preliminaries

Topics Covered

Chapter 3 (Parsing):

- Preliminaries
- Top-down Recursive Descent Parsing

Topics Covered

Chapter 3 (Parsing):

- Preliminaries
- Top-down Recursive Descent Parsing
- Top-down LL(1) Parsing

Topics Covered

Chapter 3 (Parsing):

- Preliminaries
- Top-down Recursive Descent Parsing
- Top-down LL(1) Parsing
- Bottom-up LR(1) Parsing

Topics Covered

Chapter 3 (Parsing):

- Preliminaries
- Top-down Recursive Descent Parsing
- Top-down LL(1) Parsing
- Bottom-up LR(1) Parsing
- JavaCC Parser for *j*--

Topics Covered

Topics Covered

Chapter 4 (Type Checking):

Topics Covered

Chapter 4 (Type Checking):

- Preliminaries

Topics Covered

Chapter 4 (Type Checking):

- Preliminaries
- Pre-analysis of *j*-- Programs

Topics Covered

Chapter 4 (Type Checking):

- Preliminaries
- Pre-analysis of *j*-- Programs
- Analysis of *j*-- Programs

Topics Covered

Topics Covered

Chapter 5 (JVM Code Generation):

Topics Covered

Chapter 5 (JVM Code Generation):

- Preliminaries

Topics Covered

Chapter 5 (JVM Code Generation):

- Preliminaries
- Classes and their Members

Topics Covered

Chapter 5 (JVM Code Generation):

- Preliminaries
- Classes and their Members
- Control, Message, Field Selection, and Array Access Expressions

Topics Covered

Chapter 5 (JVM Code Generation):

- Preliminaries
- Classes and their Members
- Control, Message, Field Selection, and Array Access Expressions
- Assignment, String Concatenation, Cast, and Other Operations

Topics Covered

Topics Covered

Chapter 6 (The *iota* Compiler):

Topics Covered

Chapter 6 (The *iota* Compiler):

- Preliminaries

Topics Covered

Chapter 6 (The *iota* Compiler):

- Preliminaries
- High-level Intermediate Representation (HIR)

Topics Covered

Chapter 6 (The *iota* Compiler):

- Preliminaries
- High-level Intermediate Representation (HIR)
- Low-level Intermediate Representation (LIR)

Topics Covered

Chapter 6 (The *iota* Compiler):

- Preliminaries
- High-level Intermediate Representation (HIR)
- Low-level Intermediate Representation (LIR)
- Register Allocation