Course Mechanics
Outline

1 Website
2 Goal
3 Prerequisites
4 Instructor
5 Lectures
6 Text
7 Grading
8 Software
9 Policies
10 Immediate Action Items
Website

What's on the Site?

• Announcements (landing page)
• Course Info
• Calendar
• Lecture Material
• Projects
• Resources
https://www.swamiiyer.net/cs451
Website

https://www.swamiiyer.net/cs451

What's on the Site?

• Announcements (landing page)
• Course Info
• Calendar
• Lecture Material
• Projects
• Resources
Goal

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

Practice:
• Develop a compiler (called j--) in Java for a subset language (also called j--)

Theory:

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

Practice:
- Develop a compiler (called j--) in Java for a subset language (also called j--) of Java
Prerequisites

- CS310 (Advanced Data Structures and Algorithms)
- CS420 (Intro. to the Theory of Computation)
- CS622 (Theory of Formal Languages)
- Permission of the instructor
Prerequisites

CS310 (Advanced Data Structures and Algorithms)
Prerequisites

CS310 (Advanced Data Structures and Algorithms) and

Permission of the instructor
Prerequisites

CS310 (Advanced Data Structures and Algorithms) and

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

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

Academic Interests: Evolutionary dynamics on complex networks, machine learning, programming language design, pedagogy

Other Interests: Being present, books, food, music, travel

Contact Information:
- Office: M-3-201-14
- Email: siyer@cs.umb.edu (start subject line with [CS451])

Office Hours:
- Tue Thu 10:00 AM – 12:00 PM (in-person)
- Wed 10:00 AM – 12:00 PM (remote)
Name: Swami Iyer (Senior Lecturer, Computer Science Department)
Name: Swami Iyer (Senior Lecturer, Computer Science Department)

Academic Interests: Evolutionary dynamics on complex networks, machine learning, programming language design, pedagogy
Instructor

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

Academic Interests: Evolutionary dynamics on complex networks, machine learning, programming language design, pedagogy

Other Interests: Being present, books, food, music, travel
Instructor

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

Academic Interests: Evolutionary dynamics on complex networks, machine learning, programming language design, pedagogy

Other Interests: Being present, books, food, music, travel

Contact Information:
  • Office: M-3-201-14
  • Email: siyer@cs.umb.edu (start subject line with [CS451])
Instructor

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

Academic Interests: Evolutionary dynamics on complex networks, machine learning, programming language design, pedagogy

Other Interests: Being present, books, food, music, travel

Contact Information:
- Office: M-3-201-14
- Email: siyer@cs.umb.edu (start subject line with [CS451])

Office Hours:
- Tue Thu 10:00 AM – 12:00 PM (in-person)
- Wed 10:00 AM – 12:00 PM (remote)
<table>
<thead>
<tr>
<th>Section</th>
<th>When</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TueThu</td>
<td>Y-2-2110</td>
</tr>
<tr>
<td>Section</td>
<td>When</td>
<td>Where</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>---------</td>
</tr>
<tr>
<td>1</td>
<td>Tue Thu 4:00 PM – 5:15 PM</td>
<td>Y-2-2110</td>
</tr>
</tbody>
</table>
Introduction to
Compiler Construction
in a Java World

Bill Campbell
Swami Iyer
Bahar Akbal-Delibas
Grading

Assessment % of Final Grade

Projects (1, 2, 3, 5, and best of 4 and 6) 35
Exams (1 and 2) 60
Attendance 5

Project: extensions to the \text:j--\text{ compiler
Exam: theoretical aspects of a compiler

If both exam scores $\geq 80\%$, the higher score will be the exam average

Up to $0.01\%$ extra points if $x\%$ of the class completes the end-of-semester course evaluation

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

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

If both exam scores ≥ 80%, the higher score will be the exam average. Up to 0.01% extra points if x% of the class completes the end-of-semester course evaluation. If overall score is within 0.5% of a higher grade, it will be elevated to that grade.
Grading

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

Project: extensions to the \(j--\) compiler
Grading

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

Project: extensions to the $j$-- compiler

Exam: theoretical aspects of a compiler
Grading

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

Project: extensions to the $j$-- compiler

Exam: theoretical aspects of a compiler

If both exam scores $\geq 80\%$, the higher score will be the exam average
Grading

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

Project: extensions to the $j$-- compiler

Exam: theoretical aspects of a compiler

If both exam scores $\geq 80\%$, the higher score will be the exam average

Up to $0.01x\%$ extra points if $x\%$ of the class completes the end-of-semester course evaluation
## Grading

<table>
<thead>
<tr>
<th>Assessment</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects (1, 2, 3, 5, and best of 4 and 6)</td>
<td>35</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>60</td>
</tr>
<tr>
<td>Attendance</td>
<td>5</td>
</tr>
</tbody>
</table>

Project: extensions to the $j$- compiler

Exam: theoretical aspects of a compiler

If both exam scores $\geq 80\%$, the higher score will be the exam average

Up to $0.01x\%$ extra points if $x\%$ of the class completes 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

iClicker
Software

iClicker

Piazza
Software

iClicker

Piazza

Gradescope
iClicker

Piazza

Gradescope

Programming environment
Software

iClicker

Piazza

Gradescope

Programming environment

Zoom
Policies

Classroom

Piazza

Excused Absence and Makeup Exam

Collaboration

Late Days

Regrade Request

Accommodations for students with disabilities
Policies

Classroom
Policies

Classroom

Piazza

Excused Absence and Makeup Exam
Policies

Classroom

Piazza

Excused Absence and Makeup Exam

Collaboration
Policies

Classroom

Piazza

Excused Absence and Makeup Exam

Collaboration

Late Days
Policies

Classroom

Piazza

Excused Absence and Makeup Exam

Collaboration

Late Days

Regrade Request
Policies

Classroom

Piazza

Excused Absence and Makeup Exam

Collaboration

Late Days

Regrade Request

Accommodations for students with disabilities
Immediate Action Items

- Sign up for CS account
- Sign up for iClicker
- Sign up for Piazza
- Sign up for Gradescope
- Setup the programming environment
- Fill out the questionnaire available on Gradescope
Immediate Action Items

Sign up for CS account
Immediate Action Items

Sign up for CS account

Sign up for iClicker

Sign up for Piazza

Sign up for Gradescope

Setup the programming environment

Fill out the questionnaire available on Gradescope
Immediate Action Items

Sign up for CS account

Sign up for iClicker

Sign up for Piazza
Immediate Action Items

Sign up for CS account
Sign up for iClicker
Sign up for Piazza
Sign up for Gradescope
Immediate Action Items

- Sign up for CS account
- Sign up for iClicker
- Sign up for Piazza
- Sign up for Gradescope
- Setup the programming environment
Immediate Action Items

Sign up for CS account

Sign up for iClicker

Sign up for Piazza

Sign up for Gradescope

Setup the programming environment

Fill out the questionnaire available on Gradescope