Course Mechanics
Outline

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

https://www.swamiiyer.net/cs451

What's on the Site?

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

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 $j-$) of Java
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 \textit{j--}) in Java for a subset language (also called \textit{j--}) of Java
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) and CS420 (Intro. to the Theory of Computation) or CS622 (Theory of Formal Languages); or Permission of the instructor
Prerequisites

CS310 (Advanced Data Structures and Algorithms)
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)
Prerequisites

CS310 (Advanced Data Structures and Algorithms) \textit{and} 

CS420 (Intro. to the Theory of Computation) \textit{or} CS622 (Theory of Formal Languages); \textit{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
Contact Information:
- Office: M-3-201-14
- Email: siyer@cs.umb.edu
Office Hours:
- Tue Thu 9:30 AM – 10:30 AM and 2:30 PM – 3:30 PM (in-person)
- Wed 10:00 AM – 12:00 PM (remote)
Name: Swami Iyer
Name: Swami Iyer

Contact Information:
- Office: M-3-201-14
- Email: siyer@cs.umb.edu
Name: Swami Iyer

Contact Information:
• Office: M-3-201-14
• Email: siyer@cs.umb.edu

Office Hours:
• Tue Thu 9:30 AM – 10:30 AM and 2:30 PM – 3:30 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>Tue Thu</td>
<td>4:00 PM – 5:15 PM</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>W-1-0004</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 j-- compiler
Exam: theoretical aspects of a compiler
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
<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 0.01x% 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.
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

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.
# 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

Piazza

Gradescope

Programming environment (virtual or native)

Zoom
Software

iClicker
iClicker

Piazza
Software

iClicker

Piazza

Gradescope
iClicker

Piazza

Gradescope

Programming environment (virtual or native)
Software

iClicker

Piazza

Gradescope

Programming environment (virtual or native)

Zoom
Policies

Classroom

Piazza

Collaboration

Course Staff CS451/651 Grads Classmates Others

Discuss concepts with

Acknowledge collaboration with

Expose your code/work to

View code/work of

Copy code/work from

Accommodations for students with disabilities
Policies

Classroom
Policies

Classroom

Piazza
Policies

Classroom

Piazza

Collaboration

<table>
<thead>
<tr>
<th></th>
<th>Course Staff</th>
<th>CS451/651 Grads</th>
<th>Classmates</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discuss concepts with</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Acknowledge collaboration with</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expose your code/work to</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>View code/work of</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Copy code/work from</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
Policies

Classroom

Piazza

Collaboration

<table>
<thead>
<tr>
<th></th>
<th>Course Staff</th>
<th>CS451/651 Grads</th>
<th>Classmates</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discuss concepts with</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Acknowledge collaboration with</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Expose your code/work to</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>View code/work of</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Copy code/work from</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
</tbody>
</table>

Accommodations for students with disabilities
Qualities for Success

Qualities needed to succeed in this course and as a programmer in general:

- Curiosity
- Resourcefulness
- Persistence
- Excitement
- Patience
- Concentration
- Independence
- Focus
- Creativity
- Meticulousness

*Taken from the 10 Signs You Will Suck at Programming by Jonathan Bluks*
Qualities for Success

Qualities needed to succeed in this course and as a programmer in general†:

• Curiosity
• Resourcefulness
• Persistence
• Excitement
• Patience
• Concentration
• Independence
• Focus
• Creativity
• Meticulousness

† Taken from the *10 Signs You Will Suck at Programming* by Jonathan Bluks
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
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