Contents

1 Course Information ........................................................................................................ 2
  1.1 Website ....................................................................................................................... 2
  1.2 Catalog Description ................................................................................................. 2
  1.3 Course Staff .............................................................................................................. 2
  1.4 Class .......................................................................................................................... 2
  1.5 Recommended Text .................................................................................................. 2
  1.6 Grading Scheme ......................................................................................................... 2
    1.6.1 Assessments .......................................................................................................... 2
    1.6.2 % Score to Letter Grade ....................................................................................... 3
  1.7 Software Needed ......................................................................................................... 3
    1.7.1 iClicker ............................................................................................................... 3
    1.7.2 Piazza .................................................................................................................. 3
    1.7.3 Gradescope ......................................................................................................... 3
    1.7.4 Programming Environment .................................................................................. 3
    1.7.5 Zoom ................................................................................................................... 3
  1.8 CS Account ................................................................................................................ 3
  1.9 Policies ....................................................................................................................... 3
    1.9.1 Classroom ............................................................................................................. 3
    1.9.2 Piazza .................................................................................................................. 4
    1.9.3 Makeup Exam ...................................................................................................... 4
    1.9.4 Late Days ............................................................................................................. 4
    1.9.5 Regrade Request ................................................................................................. 4
    1.9.6 Collaboration ....................................................................................................... 4
    1.9.7 Accomodations for Students with Disabilities ...................................................... 4
    1.9.8 Campus Closure ................................................................................................... 4

2 Topics Covered .................................................................................................................. 4

3 Assignments ..................................................................................................................... 5
  3.1 The List ....................................................................................................................... 5
  3.2 Submitting Your Work ............................................................................................... 6
  3.3 How the Assignments will be Scored ......................................................................... 6
    3.3.1 Correctness .......................................................................................................... 6
    3.3.2 Clarity and Efficiency ......................................................................................... 6
    3.3.3 Notes File .......................................................................................................... 6
1 Course Information

1.1 Website

https://www.swamiiyer.net/cs451/

1.2 Catalog Description

Introduction to compiler organization and implementation, including formal specifications and algorithms for lexical and syntactic analysis, internal representation of the source program, semantic analysis, run-time environment issues and code generation. Students will write a compiler for a reasonably large subset of a contemporary language, targeted to a virtual machine.

Prerequisites: CS310 and CS420 or CS622; or permission of the instructor.

Students who successfully complete this course will be able to: write parsers and produce an abstract syntax tree (AST); analyze and generate code for a programming construct represented by an AST; and allocate physical registers (a limited resource) to a program expressed in terms of virtual registers (an unlimited resource).

1.3 Course Staff

Swami Iyer will be the primary instructor for the course. He will be assisted by a graduate teaching assistant (TA).

1.4 Class

There will be two 75-minute classes per week. In each class, the instructor will present the material for that class for an hour, and conduct an online quiz on that material for the remaining 15 minutes. You are strongly encouraged to review the material before each class, and review it again thoroughly soon after.

1.5 Recommended Text

*Introduction to Compiler Construction in a Java World* by Bill Campbell, Swami Iyer, and Bahar Akbal-Delibaš

This text enables a deep understanding of the Java programming language and its implementation. It covers all of the standard compiler topics, including lexical analysis, parsing, abstract syntax trees, semantic analysis, code generation, and register allocation.

1.6 Grading Scheme

1.6.1 Assessments

<table>
<thead>
<tr>
<th>Item</th>
<th>% of Final Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Assignments (best 4 of 1, 2, 3, 4, and 6; and 5)</td>
<td>40</td>
</tr>
<tr>
<td>Exams (1 and 2)</td>
<td>50</td>
</tr>
<tr>
<td>Quizzes (5 lowest scores will be dropped)</td>
<td>10</td>
</tr>
</tbody>
</table>

- The goal of the programming assignments is to make sure that you can apply the concepts learned in class to enhance the functionality of the base *j* and *iota* compilers.
- The closed-book exams will test your understanding of the theoretical concepts covered in class.
The online quiz conducted at the end of each class will assess your understanding of the material covered in that class. Each question in a quiz is worth 2 points (1 point for correct response and 1 point for responding). Each quiz score will be normalized to 100 points.

- You can earn 0.01x% extra points if x% of the class completes the end-of-semester course evaluation.
- If your overall score falls within half a percent of a higher grade, your score will be elevated to that grade.

1.6.2 % Score to Letter Grade

\[
\begin{align*}
[93, 100): & \text{ A}, \\
[90, 93): & \text{ A-}, \\
[87, 90): & \text{ B+}, \\
[83, 87): & \text{ B}, \\
[80, 83): & \text{ B-}, \\
[77, 80): & \text{ C+}, \\
[73, 77): & \text{ C}, \\
[70, 73): & \text{ C-}, \\
[67, 70): & \text{ D+}, \\
[63, 67): & \text{ D}, \\
[60, 63): & \text{ D-}, \\
[0, 60): & \text{ F}
\end{align*}
\]

1.7 Software Needed

1.7.1 iClicker

We will use iClicker to conduct in-class quizzes, for which you will need an iClicker Student App Subscription.

1.7.2 Piazza

We will use Piazza as the Q&A platform for the course. If you have any general questions about the assignments, exams, or the lecture material, the most effective way to get them answered is by posting them on Piazza. You can expect your questions to be answered by the course staff or your peers.

1.7.3 Gradescope

We will use Gradescope to grade your programming assignments and exams.

1.7.4 Programming Environment

To write and execute Java programs in this course, you will need a laptop (Linux, Mac, or Windows) properly configured with the necessary software. Click here for setup instructions.

1.7.5 Zoom

We will use Zoom to hold remote office hours.

1.8 CS Account

In order to use the computing resources of the department, and in particular, those in the UNIX/PC Lab (M-3-0731), you need to setup a CS account. With your CS account credentials, you can connect to our designated server (users.cs.umb.edu) remotely using SSH. With the same credentials, you can also sign into the Linux systems in the CS Lab. In addition, you can sign into the Windows systems in the lab with the same username and an initial password abcd_1234, which you must change the first time you sign in.

Visit CS Labs Portal to register for a portal/CS account and confirm via email. If you already have a CS account, use the same username. The next step is to sign into the portal and select your courses for the term. You will be notified via your UMB email once the course directories and your account are created.

1.9 Policies

1.9.1 Classroom

Come to class on time and stay for the entire session. If you have to leave early, let the instructor know in advance. Have your mobile phone silenced or turned off during the entire session. Use of earphone/headphone during the session is not permitted. Use of a laptop during the session is permitted only for class purposes. Do not talk to each other during the session. If you have any questions, bring them up to the instructor.
1.9.2 Piazza

If you have a question, first make sure that it has not already been asked/answered. Clearer questions get better answers, so re-read your question before you post it. Ask your questions early. Posts are categorized using channels, so pick an appropriate tag for your post. Use the platform only for questions that can be asked in a general way, without sharing code or other assignment-related work. However, if you are stuck on a problem despite your valiant efforts to solve it, you may seek help from the course staff by posting your code privately, as properly formatted text (not images). Any post that is inappropriate or violates the academic honesty code will be deleted by the course staff.

1.9.3 Makeup Exam

You must provide appropriate documentation if you were/are unable to take an exam on the scheduled date and want to arrange a makeup exam. The documentation must be a letter from the Dean of Students if the type of your absence is among those listed on their website. For other types of absences, the supporting documentation must be emailed to the instructor directly.

Note: There will be no makeup for missed quizzes.

1.9.4 Late Days

Assignment deadlines are firm and non negotiable, but you have up to 5 (8 if you have accommodations through the Ross Center) late days that you may use to submit one or more assignments a bit late. You may request an extension for any assignment only once. To do so, you must send the instructor a request via email at least 48 hours prior to the assignment deadline, clearly stating the number of late days needed, or else your request will not be granted.

Unused late days will be converted into \( x\% \) extra points, which will be added to your final grade. \( x \) will be calculated as \( x = \frac{d}{n} \), where \( n \) is the number of late days allotted and \( d \) is the number of late days unused. For example, if \( d = 3 \) and \( n = 5 \), then \( x = 0.6\% \).

1.9.5 Regrade Request

If you have any concerns about the grading of a particular assignment or exam, you may submit a regrade request via Gradescope. You must submit the request within a week from the date the assignment or exam grades are published, or else your request will be turned down.

1.9.6 Collaboration

Click here for the collaboration policy and the penalties for infractions of the policy.

1.9.7 Accomodations for Students with Disabilities

Section 504 of the Americans with Disabilities Act of 1990 offers guidelines for curriculum modifications and adaptations for students with documented disabilities. If applicable, students may obtain adaptation recommendations from the Ross Center for Disability Services. The student must present these recommendations and discuss them with the instructor within a reasonable period, preferably by the end of Add/Drop period.

1.9.8 Campus Closure

In the event of a campus closure, all class-related activities will be conducted remotely, via Zoom. If there is an exam scheduled to take place on that day, the exam will be postponed to the next suitable date.

2 Topics Covered

- Course Mechanics [Lecture 1]
- Programming Environment [Lecture 1]
• Chapter 1: Compilation
  – Preliminaries [Lecture 2]
  – Overview of the j-- to JVM Compiler [Lecture 3]
  – Overview of the iota to Marvin Compiler [Lecture 4]

• Chapter 2: Scanning
  – Preliminaries [Lecture 5]
  – Handcrafting a Scanner [Lecture 6]
  – Generating a Scanner [Lecture 7]
  – JavaCC Scanner for j-- [Lecture 8]

• Chapter 3: Parsing
  – Preliminaries [Lecture 9]
  – Top-down Recursive Descent Parsing [Lecture 10]
  – Top-down LL(1) Parsing [Lecture 11]
  – Bottom-up LR(1) Parsing [Lecture 12]
  – JavaCC Parser for j-- [Lecture 13]

• Chapter 4: Type Checking
  – Preliminaries [Lecture 14]
  – Pre-analysis of j-- Programs [Lecture 15]
  – Analysis of j-- Programs [Lecture 16 and 17]

• Chapter 5: JVM Code Generation
  – Preliminaries [Lecture 18]
  – Classes and their Members [Lecture 19]
  – Control, Message, Field Selection, and Array Access Expressions [Lecture 20]
  – Assignment, String Concatenation, Cast, and Other Operations [Lecture 21]

• Chapter 6: Marvin Code Generation
  – Preliminaries [Lecture 22]
  – High-level Intermediate Representation (HIR) [Lecture 23 and 24]
  – Low-level Intermediate Representation (LIR) [Lecture 25]
  – Register Allocation [Lecture 26 and 27]

3 Assignments

3.1 The List
There are 6 programming assignments in all. These are due at midnight (11:59 PM to be precise) on the dates indicated on the Calendar page of the course website.
# Title Goal

1 Supporting Simple Operations Become familiar with the Java Virtual Machine (JVM) and the Marvin Machine; and extend the \textit{j--} language by adding support for some arithmetic operators, conditional expression, and \textit{do} statement.

2 Scanning Modify the handcrafted scanner to support multiline comments; additional tokens (reserved words and operators); and \texttt{long} and \texttt{double} literals in \textit{j--}.

3 Parsing Modify the handcrafted parser to support \texttt{long} and \texttt{double} basic types; additional operators; for, break, continue, and switch statements; exception handlers; and interface type declaration in \textit{j--}.

4 Scanning and Parsing with JavaCC Modify the \texttt{j--.jj} file used for generating a scanner and parser (using \texttt{JavaCC}) for \textit{j--} to support multiline comments; \texttt{long} and \texttt{double} basic types; additional tokens (reserved words and operators); conditional expression; do, for, break, continue, and switch statements; exception handlers; and interface type declaration.

5 Type Checking and Code Generation Implement type checking and JVM code generation for the programming constructs that were added to \textit{j--} in Assignment 3 (Parsing).

6 Register Allocation Implement the graph coloring based register allocation algorithm in \texttt{iota}.

## 3.2 Submitting Your Work

You will use Gradescope to submit your Java programs (ie, .java files) and the notes.txt file. Make sure that you only submit files listed under the “Files to Submit” section of the assignment writeup.

You may submit your files as many times as you like, up until the assignment deadline. The most recent submission is considered active by default and your score on the active submission is your official score for the assignment as well. You have the option of making any of your previous submissions active.

Note: If your active submission is partial, your assignment score will also be partial, so in order to be eligible for full credit, make sure you have an active submission containing all the required files for the assignment.

## 3.3 How the Assignments will be Scored

### 3.3.1 Correctness

Your solution to each assignment problem will be evaluated for correctness by an autograder. Each test that is used for this purpose is worth some number of points; your solution will receive all the points from a test that passes and 0 points from a test that does not pass. Your overall correctness score will be normalized to 80 points.

### 3.3.2 Clarity and Efficiency

Your solution to each assignment problem will additionally be checked by a TA for clarity and efficiency. Your code will receive 10 points if it meets our expectations (must include adequate comments, must follow good programming principles, and must meet any problem-specific requirements), and will be marked down otherwise.

### 3.3.3 Notes File

The given notes.txt file for an assignment must be uploaded with the three sections (#1 mandatory, #2 if applicable, and #3 optional) filled in as appropriate. Your notes file will receive 10 points if it meets our expectations (section #1 must provide a clear high-level description of each problem in no more than 100 words), and will be marked down otherwise.