COURSE OUTLINE

CS 4110
Concepts of Formal Languages and Algorithms for Computing
Spring Semester 2013
Weber State University

ADMINISTRATION

Instructor: Dr. Brian Rague
Email: brague@weber.edu
Web Page: http://faculty.weber.edu/brague
Phone: 626-7377
Office Location: TE110-F
Office Hours: MW from 9:30am-11:30am, T from 10:30am-11:30am

Time and Room: 7:30 – 9:20 MW in TE109C (with Lab)

Text: Introduction to Computer Theory (2nd ed.) by Daniel I.A. Cohen

Pre-requisites: CS2420 and MA1630/CS3130

COURSE DESCRIPTION

An overview of the concepts of formal language definition, automata theory, Turing theory, and solvability, and an introduction to algorithms and computational methods used in advanced computer science courses.

CLASS PROCEDURE

This class focuses heavily on understanding the conceptual foundations that govern the interaction between languages and machines. Since applying these ideas in a practical context is also emphasized, the course will have a programming lab component. To foster mastery of the course material, the class will have both individual and group activities during supervised class and outside class times.

We will cover the key areas of modeling formal languages through regular expressions, automata, and grammars. There will be ample opportunity to explore all the facets of formal languages and algorithms in lab work and in-class quizzes and exercises. Classes will meet each week at the assigned hour. Class discussions will typically take place during the first half of each class, at which time topics will be presented and questions about the reading, assignment, and labs are welcomed at this time. Exercises will be reviewed. Class will consist of lecture and discussion on topics assigned through this outline and class assignments. We will cover the elements of computational modeling in both its theoretical and practical contexts. Both individual programming assignments and group quizzes and exercises can be worked on during specified lab periods typically provided during the last half of the class. The due dates for the lab will be confirmed when the lab is assigned.
On any given class meeting of roughly 1 hour and 50 minutes, the first half will be dedicated to traditional presentation and discussion focused on primary subject matter concepts. Topics addressed during these class presentations will be driven by student inquiry and the results of the in-class assessments. The second half of every class meeting will be devoted to the individual/group assessments, individual homework, or group exercises depending on the current needs of the class.

COURSE GOALS

- To become familiar with the mathematical models of computers and formal language that will provide a foundation for understanding the limitations of computing.
- To provide the background needed by those students who decide to continue on to graduate studies in Computer Science.
- To develop skills in producing mathematical proofs about models of programming and formal languages.
- To understand the machine and language models associated with regular, context-free, and phrase-structure languages.
- To understand the issues of decidability, the Halting Problem, Church’s Thesis, and computability theory.

GENERAL EXPECTATIONS

Attendance

CS 4110 covers a broad range of information that requires some form of interaction and participation in the classroom setting. A portion of the final grade includes in-class assessments and group exercises, which is greatly influenced by a combination of attendance and input during class discussions. It is vital that you attend all scheduled classes and labs. Your punctuality and attentiveness (not reading newspapers, not surfing the web) is appreciated and will be noted. Activated cell phones, because of their capacity to generate noises and to be spoken into, are not allowed. Students are responsible for material covered during absences regardless of the reason for the absence.

Academic Honesty

Academic honesty is highly valued at Weber State University and within this class. A student must always submit work that represents his or her original words or ideas. If any words or ideas are used that do not represent the student's original words or ideas, the student must cite all relevant sources. The student should also make clear the extent to which such sources were used. Words or ideas that require citations include, but are not limited to, all hardcopy or electronic publications, whether copyrighted or not, and all verbal or visual communication when the content of such communication clearly originates from an identifiable source.

Individuals involved in any acts of cheating or plagiarism will be given a failing grade for the course. In addition, names of these individuals will be submitted for disciplinary action by the department and the university.

Academic dishonesty involves any and all of the following:

- Having a tutor or friend complete a portion of your assignments
• Having a reviewer make extensive revisions to an assignment
• Copying work directly from another student
• Using information from online information services without proper citation

Team programming is not discouraged but any collaboration should be at the conceptual level. If detail source code is duplicated from one program into another, it will be deemed as cheating and severe action will be taken against persons who knowingly have participated. Source code that doesn't match the output results may also be deemed as cheating which may result in severe penalties.

Students are expected to be familiar with the WSU Student Code and abide by it. The Code may be reviewed online at http://www.weber.edu/ppm/6-22.htm (pay specific attention to Section 4D). All necessary steps will be taken to enforce the Student Code to guarantee fairness to all students.

CS Department policy dictates that any verifiable evidence of student academic cheating, as defined and determined by the instructor, will result in: 1) an automatic failing grade for the class and 2) a report to the Dean of Students that will include the student's name and a description of the student's dishonest conduct.

**Readings, Assignments, & Labs**

As you may already know, reading a textbook about computer theory is not like reading a novel. Because of the focus and concentration required to understand the material, even one page of reading for this course could require upwards to an hour of your time. Manage your time accordingly for all weekly reading assignments, which should be completed prior to classroom instruction. Because in-class assessments are based primarily on material in the course text, *it is highly recommended that you complete the assigned weekly readings prior to taking the assessment.* There are 3 individual programming assignments. **The individual programs will be graded on proper documentation, successful compilation, execution, and satisfaction of requirement specifications as given in the assignment description.** Due dates for individual assignments will be determined during class. Each student will upload assignments to a designated class server, typically using the current WSU Online portal. The uploading procedure will be explained during the first week of the term. **Assignments may not be submitted by email.**

The computing labs and math tutors are available to help you complete your assignment. I will be present during most labs scheduled during class time to assist you. Use this lab time to explore computing tools, collaborate with fellow team members, and get some hands-on experience with the concepts discussed in class.

**Programming Software**

As part of this course, you will be designing and constructing programs. **All imperative programs should be written in either Java or C/C++.** You will be directed toward compiler resources and programming environments as the need arises. These can include any of the following: Visual Studio, Gnu C Compiler, Java Development Kit (JDK), Eclipse, Borland C++ Builder, etc. Try to introduce yourself to a Java and/or C programming package as early as possible.
The WSU CS department has an Academic Alliance program with Microsoft that allows students to obtain Microsoft Visual Studio for free. The student chapter of ACM handles the ordering and distribution of this software.

The website for the MSDNAA is [http://msdnaa.cs.weber.edu/](http://msdnaa.cs.weber.edu/). If DNS issues prevent the site from being visible outside of the main campus labs, alternatively you can try [http://137.190.19.14](http://137.190.19.14).

You can also find ANSI standard C++ compilers on the web for free. Any ANSI compliant compiler should work. 'Bloodshed' is an aptly named example. If you find one that works well, please post so that others may benefit.

**Important Information about Your Programs and Assignments.**

Programming assignments should be uploaded using the current WSU Online portal assigned to this class. Details will be given during class. Typically only the source code (.cpp, .java file) should be submitted. Executables are rarely required. I will compile the submitted source in order to confirm that no syntax errors exist within the code. **All submitted programs should compile as a minimum requirement.**

Avoid submitting programs by email, because there’s always a distinct possibility I may not receive the assignment due to some obscure networking, server, or mail application problem. I will not confirm receipt of assignments emailed to me.

Collaboration on *individual* assignments should be at the conceptual level only. If detail source code is duplicated from one program into another, it will be deemed as cheating and severe action will be taken against persons who knowingly have participated. Students should ensure that the latest version of source code is submitted. Source code that doesn't match the output results may also be deemed as cheating which may result in severe penalties.

**Documenting Your Programs**

Program documentation is very important. For CS4110, we will emphasize explicit documentation for the individual programs, as indicated in the separate section below.

**I. Individual code**

Program documentation for individual assignments will consist of both prologue and explanatory comments. Please note the following three requirements regarding the documentation for your programs.

1. **The comment headers for each of your submitted programs must adhere to the following format:**

```plaintext
// John Doe
// CS 4110 - 7 am
// Assignment #99
// Dr. Rague
// Due: mm/dd/yy
// Version: 1.0
// --------------------------------------------------------
```
This program calculates the revenue generated by an employee installing coaxial cable

---

```
#include <iostream>
using namespace std;

2. Each main modular unit (classes and functions) should be preceded by a short description as shown here:

---

This class represents a circle that can be displayed on a console window.

---

class Circle
{
    protected:
        int x, y;

3. You should also include appropriate in-line comments throughout your code to clarify any sections or statements in which the programming strategy isn't immediately obvious.

---

int Num_Cases;            // The number of cases of data values to process
int Installations;        // The number of installations
double Yards_Of_Cable,    // The yards of cable installed
    Feet_Of_Cable,    // The number of yards of cable converted to feet
    Revenue;        // The revenue generated for this case of data

---

Late Assignment Policy

Please note that a late assignment will be accepted only within the first five days following its original due date. The maximum grade possible for a late assignment will be automatically reduced by an amount equal to 10% of the original total point value for each day it is late, and, as stipulated, will not be accepted at all after five days. No assignments will be accepted after the last day of class, prior to finals week. No exceptions.

“I really, really need to get a C” Policy:

The most effective method for obtaining a C or above in this class is to submit assignments when they are due and to stay current with course topics. The curriculum is carefully designed to fit the number of course weeks. In order to uphold academic rigor and integrity, student grades must be based on the degree to which the course requirements listed in the syllabus are fulfilled. Except when specifically made available per the course schedule, extra credit
assignments are not allowed. If you approach me anytime during the term claiming that special allowance should be made because you need a C to move forward in the program, graduate, receive financial aid, etc., I will decline your request and refer you to this clearly worded policy.

**Exams & Quizzes**

There will be eight individual/group in-class assessments, and three exams. The in-class assessments will be composed of multiple choice, short answer, and true/false questions. These are primarily used to test your knowledge of key terminology, operations, and computational concepts that have been introduced in class presentations. Each student will take the test individually, and then each team will take the same test as a group. These assessments are primarily used to test your knowledge of key terminology and programming concepts that have been provided in the reading and highlighted in the class presentations. To receive credit, each student must take the in-class assessment both individually and as a team member. As long as you read and attend class sessions, you should have no problem receiving a good score on the tests.

- Assessment 1 – Week 2
- Assessment 2 – Week 4
- Assessment 3 – Week 6
- Assessment 4 – Week 7
- Assessment 5 – Week 9
- Assessment 6 – Week 11
- Assessment 7 – Week 12
- Assessment 8 – Week 14

- Exam 1 – Week 5
- Exam 2 – Week 10
- Exam 3 – After Week 14

The exams will be given at a specific testing site on campus. Information about how the exams will be administered will be given during the week before the exam.

*The exams are not to be shared or discussed with other students. If I find out or discover that there has been any cheating, it will result in an automatic failure for the course!*

Each student will receive the grade, which in the best judgment of the instructor, he or she has earned. Requests for a grade higher than the one earned, for any reason (including loss of student visa, entrance requirements into a professional school, etc.) will not be honored. There will be no retests. Make-up exams will be given only for valid reasons and if the instructor is notified in advance of the scheduled exam hour.

**Grading Policy**

- Small Team In-Class Exercises: 20%
- Individual Programming Assignments: 20%
- Individual/Group Assessments: 15%
<table>
<thead>
<tr>
<th>LETTER GRADE</th>
<th>PERCENT RANGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>total &gt;= 93.3%</td>
</tr>
<tr>
<td>A-</td>
<td>90.0% &lt;= total &lt; 93.3%</td>
</tr>
<tr>
<td>B+</td>
<td>86.7% &lt;= total &lt; 90.0%</td>
</tr>
<tr>
<td>B</td>
<td>83.3% &lt;= total &lt; 86.7%</td>
</tr>
<tr>
<td>B-</td>
<td>80.0% &lt;= total &lt; 83.3%</td>
</tr>
<tr>
<td>C+</td>
<td>76.7% &lt;= total &lt; 80.0%</td>
</tr>
<tr>
<td>C</td>
<td>73.3% &lt;= total &lt; 76.7%</td>
</tr>
<tr>
<td>C-</td>
<td>70.0% &lt;= total &lt; 73.3%</td>
</tr>
<tr>
<td>D+</td>
<td>66.7% &lt;= total &lt; 70.0%</td>
</tr>
<tr>
<td>D</td>
<td>63.3% &lt;= total &lt; 66.7%</td>
</tr>
<tr>
<td>D-</td>
<td>60.0% &lt;= total &lt; 63.3%</td>
</tr>
<tr>
<td>F</td>
<td>total &lt; 60.0%</td>
</tr>
</tbody>
</table>

Students with Disabilities

Any student requiring accommodations or services due to a disability must contact Services for Students with Disabilities (SSD) in room 181 of the Student Services Center. SSD can also arrange to provide course materials (including the syllabus) in alternative formats if necessary.

For more information about the SSD contact them at 801-626-6413, ssd@weber.edu, or departments.weber.edu/ssd.

WSU Student Code

Students are expected to be familiar with the WSU Student Code and abide by it. The Code may be reviewed online at http://www.weber.edu/ppm/6-22.htm (pay specific attention to Section 4D). All necessary steps will be taken to enforce the Student Code to guarantee fairness to all students.

Bad Weather Policy

Use good judgement in the event of bad weather. Do not take unnecessary risks. The attendance and participation policies will be relaxed in the event of inclement weather. Exams will be rescheduled as necessary. Keep informed about class cancellations by listening to the radio.
**Disclaimer**

The following syllabus is subject to change at any time. The listing represents a *tentative* class schedule for the term and the topics covered. Alterations made in class supersede this document.

**CS 4110 Syllabus:**

<table>
<thead>
<tr>
<th>WEEK</th>
<th>STARTING</th>
<th>TOPICS/PRACTICE HOMEWORK EXER.</th>
<th>READING</th>
<th>WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Monday, Jan 7</td>
<td>Introduction, Background, Languages</td>
<td>Cohen – Chapters 1 &amp; 2</td>
<td><em>Schedule:</em> Mon: Groups Formed, Practice Assessment</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 2    | Monday, Jan 14 | Recursive Definitions, Regular Expressions                          | Cohen – Chapters 3 & 4 | *Schedule:* Mon: In-Class Exercise w/Extra Credit  
|      |                | Chapter 2: 5, 8, 11  
|      |                | Chapter 3: 15, 19  
|      |                | Chapter 4: 12, 18                                                  |               | Wed: Assessment #1 on Week 1 & 2 Readings |
| 3    | Monday, Jan 21 | Finite Automata, Transition Graphs                                  | Cohen – Chapters 5 & 6 | *Schedule:* Mon: In-Class Exercise w/Extra Credit  
|      |                | Chapter 5: 14, 17  
|      |                | Chapter 6: 2, 16                                                   |               | Assignments: Programming Assignment #1 –  
|      |                |                                                                     |               | Due Wednesday, midnight |
| 4    | Monday, Jan 28 | Kleene’s Theorem, Finite Automata with Output, Regular Languages    | Cohen – Chapters 7, 8 & 9 | *Schedule:* Mon: In-Class Exercise w/Extra Credit  
|      |                | Chapter 7: 1  
|      |                | Chapter 8: 1, 9  
|      |                | Chapter 9: 6                                                      |               | Wed: Assessment #2 on Week 3 & 4 Readings |
| 5    | Monday, Feb 4  | Nonregular Languages, Decidability                                 | Cohen – Chapters 10 & 11 | *Schedule:* Mon: In-Class Exercise w/Extra Credit  
|      |                | Chapter 10: 4, 16  
|      |                | Chapter 11: 2, 8, 13                                               |               | Wed: Exam Review |
|      |                |                                                                     |               | *Exam 1* on material covered from Week 1 up through and including Week 5. |
| 6    | Monday, Feb 11 | Context-Free Grammars, Grammatical Format                           | Cohen – Chapters 12 & 13 | *Schedule:* Mon: In-Class Exercise w/Extra Credit  
|      |                |                                                                     |               | Wed: Assessment #3 on |


<table>
<thead>
<tr>
<th>Week of Mar 4th – Spring Break</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Cohen – Chapters 16 &amp; 17</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Cohen – Chapters 19 &amp; 20</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Cohen – Chapters 21 &amp; 22</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Date</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>14</td>
</tr>
</tbody>
</table>

**Schedule:**
- **Mon:** Assessment #8 on Week 13 & 14 Readings
- **Wed:** Exam Review

**Exam 3** on material covered from Week 11 up through and including Week 14.