COSC 461: Compilers
General Information – Fall 2017

Class Information: As below:

Room: 404 Min H. Kao Building
Time: MWF 3:35PM – 4:25PM
Prerequisites: COSC 302 (or COSC 307): Data Structures and Algorithms, COSC 312: Algorithm Analysis and Automata (recommended)

Instructor: Michael Jantz

Office: 607 Min Kao (Ph: 865-974-5470)
Office Hours: F 1:30PM – 3:30PM or by appointment
Email: mrjantz@utk.edu

Grader: Divyani Rao

Office: Min Kao 636
Office Hours: TBD or by appointment
Email: drao@vols.utk.edu

Texts: Aho, Lam, Sethi, and Ullman. Compilers: Principles, Techniques, and Tools (2nd Edition). Recommended: Harbison and Steele. C: A Reference Manual.

Course Website: The course will be hosted on UTK's Blackboard site: https://bblearn.utk.edu/. The page will contain a variety of information, which will include the syllabus, schedule, slides, assignments, and grades.

Slides: There is a lot of material to cover in this class. Lecturing from slides will allow me to cover the material at a more rapid pace. I will be presenting slides that I have developed and slides of figures and tables from the text. Slides and additional material that I have developed for the class will be made available from the course website prior to their presentation.

Grading: Grades will be based on your scores over two midterms (15% each) and final exam (20%), programming assignments (40% total), and quizzes and participation (10%). Keep all graded material to provide evidence of grades. I plan to use a standard grading scale (90 to 100 is an A, 80 to 89 is a B, and so on.), but I reserve the right to alter this scale depending on how the class scores on tests and assignments.

Attendance and Punctuality: I do not plan to take roll, but you are responsible for all material presented in class. If attendance becomes an issue, I reserve the right to check roll to evaluate participation grades. Exams, quizzes, and due dates will be scheduled in advance. A grade of zero will be recorded for missed exams and late assignments unless prior arrangements are made. Assignments turned in after the due date, but by the beginning of the next scheduled class will be penalized 10%. Assignments will not be accepted that are more than one class period late.

Cheating: Students are encouraged to discuss programs in general and to help one another find bugs in existing programs. Copying another's code or writing code for someone else is cheating. I will fail you if you cheat. Please review your student handbook for additional details on what constitutes academic misconduct.

Disabilities: Please advise the instructor of this class at your earliest convenience (minimum of five working days) if you have a disability that will require a reasonable accommodation for any of the activities in the course schedule.

Programming Assignments: There will be five major programming assignments. In some cases, successive assignments will build on previous assignments. However, inability to successfully complete a particular assignment or phase is no reason to panic. Executables or code for previous phases will be provided when necessary. The following list shows the assignments the language you will use for your implementation, and the weight each assignment will count towards your grade.

1. NFA2DFA converter – your choice of language, our solution uses Python (5%)
2. lexical converter – using lex (10%)
3. calculator with scanner/parser tools – lex and yacc (10%)
4. csem C semantic routines – C (15%)