

CSCI450: Compiler Design

1.	Course Prefix and number	CSCI 450
2.	Number of semester hours	3
3.	Pre-requisites	CSCI 250 and CSCI 330 as a corequisite.
4.	Location of classroom and time course meets	WS 118 2:00-3:15 Tue, Thu
5.	Intstructor Web Page	Dr. Warren MacEvoy http://www.coloradomesa.edu/~wmacevoy
6.	Office Hours	See my web page.
7.	Course Description	Structures and techniques used in compiler writing are discussed with emphasis on scanners, symbol tables, parsers and code generation. The front end of a recursive descent parser is written for the semester project. Error analysis and code optimization are discussed as time permits.
8.	Essential Learning Objectives	This is not an essential learning class.
9.	Textbook and Materials	<u>Modern Compiler Design</u> , by Grune, et al. This is available as a free eBook through the campus library.
10.	Course Objectives	Learning the theory and practice of accomplishing what is in the course description [:o)]. (CS Course Objectives)
11.	Policy on absences and tardies	Make up exams will only be given with advanced notification or very unusual circumstances. You are responsible for material given in class, including changes of schedule for exams.

12.	Policy on late work	Late projects cost 2 letter grades per day.
13.	Policy on academic dishonesty	You may be given a failing grade on a project or the course overall for academic dishonesty.
14.	Disability Arrangements	In coordination with Educational Access Services, reasonable accommodations will be provided for qualified students with disabilities. Please meet with the instructor the first week of class to make arrangements. Nancy Conklin, the Coordinator of Educational Access Services, can be contacted at 248-1826, or in person in Houston Hall, Suite 108.
15.	Evaluation and grading	One-third of the grade will reflect performance on assigned individual projects, one-third of the grade will be performance on steps of the group project, and one-third will be based on the final project. Projects will be graded on an A, B, or F basis: • A: Well written and documented, with more bells and whistles than required for the project. Testing procedures must be included with the project. • B: Works correctly. Reasonably written. Reasonably documented. Reasonably tested. • F: Fails to work. Poorly written, documented, or tested.
16.	Major assignments	The most important assignment will be the development of a compiler and/or interpreter.
17.	Course Topics	See below

Week	Tue	Thu
17-Aug	Tue 19-Aug 1	Thu 21-Aug 2
24-Aug	Tue 26-Aug 3	Thu 28-Aug 4
31-Aug	Tue 02-Sep 5	Thu 04-Sep 6
7-Sep	Tue 09-Sep 7	Thu 11-Sep 8
14-Sep	Tue 16-Sep	Thu 18-Sep

	9	10
21-Sep	Tue 23-Sep 11	Thu 25-Sep 12
28-Sep	Tue 30-Sep 13	Thu 02-Oct 14
5-Oct	Tue 07-Oct 15	Thu 09-Oct 16
12-Oct	Tue 14-Oct 17	Thu 16-Oct 18
19-Oct	Tue 21-Oct 19	Thu 23-Oct 20
26-Oct	Tue 28-Oct 21	Thu 30-Oct 22
2-Nov	Tue 04-Nov 23	Thu 06-Nov 24
9-Nov	Tue 11-Nov 25	Thu 13-Nov 26
16-Nov	Tue 18-Nov 27	Thu 20-Nov 28
23-Nov	Tue 25-Nov *Thanksgiving*	Thu 27-Nov *Thanksgiving*
30-Nov	Tue 02-Dec 29	Thu 04-Dec 30
7-Dec	Tue 09-Dec *Finals*	Thu 11-Dec *Finals*

TOPIC	IDEA	Section
1	intro	
2	syntax and translation	
3	parsing	
4	example	
5	lexing	
6	symbol table	
7	intermediate code	
8	test	

9	lexical analysis	
10	buffering	
11	tokens	
12	lex	
13	automata	
14	re -> fa	
15	syntax	
16	grammars	
17	top-down parsing	
18	bottom-up parsing	
19	ambiguity	
20	generators	
21	syntax directed translation	
22	syntax directed application	
23	schemes	
24	intermediate code	
25	types	
26	expressions	
27	type check	
28	flow control	
29	procedures	
30	storage	
31	non-local access	

32	heap and gc	
----	-------------	--

Project # 1: UTF8 decoder. Batch translate raw UTF8 encoded file into sequence of UNICODE Code Points:

Ex: File has "H llo" program outputs

72

399

108

108

111

10 (maybe)