CPCS-302 Syllabus

Catalog Description

CPCS-302 Compiler Construction
Credit: 3 (Theory: 3, Lab: 0, Practical: 1)
Prerequisite: CPCS-301
Classification: Department Required

The objective of this course is to acquaint students with the fundamentals of compilers and their construction. The course considers the principles that underlie compiler construction and focuses on the translation of programs written in conventional, higher level language into semantically equivalent programs written in assembly language. Students will learn how modern programming languages are implemented, how compilers interact with operating systems and machine architecture, and how to use compiler construction tools.

Class Schedule

Lab/Tutorial 90 minutes 1 times/week

Meet 50 minutes 3 times/week or 80 minutes 2 times/week

Textbook

Alfred V. Aho, , "Compilers", Addison Wesley Publishing Company; 2 edition (2007)

ISBN-13 9780321486813 **ISBN-10** 0321486811

Grade Distribution

Week	Assessment	Grade %
7	Graded Lab Work 1	2
7	Exam 1	15
10	Quiz	5
12	Exam 2	20
12	Graded Lab Work 2	3
15	Group Project	20
15	Lab Exam	5
16	Comprehensive Final Exam	30

Last Articulated

April 3, 2018

Relationship to Student Outcomes

a	b	c	d	e	f	g	h	i	j	k
x	x	x						x		

Course Learning Outcomes (CLO)

By completion of the course the students should be able to

1. Define different compiler phases (i)
2. Recognize the relationship between compiler and other computer software programs. (i)
3. Analyse regular expressions to its tokens: keywords, operators, identifiers, and expressions. (a)
4. Express a grammar in derivations and parse tree. (a)
5. Convert regular expression to its equivalent automata (i)
6. Design a deterministic finite state machine to accept a specified language. (i)
7. **Produce parse tree from context - free grammar. (a)**
8. Identify top-down and bottom-up parsing. (b)
9. Build a recursive descent parser for a mini language. (c)
10. **Construct LL(1) parsing table. (i)**
11. Construct simple LR parsing table. (i)
12. **Construct different types of LR parsing table. (i)**
13. **Use lexical analyser and parser generator tools: lex, YACC, and JavaCC. (i)**
14. Discuss syntax-directed techniques for translation. (a)
15. **Represent intermediate code using two and three-address schemes. (b)**

Coordinator(s)

Prof. Imtiaz Khan, Professor

Prof. Manal Abdullah, Professor

CPCS-302 Syllabus

Topics Coverage Durations

Topics	Weeks
Introduction to Compilers	1
The Structure of a Compiler	1
A Simple Syntax-Directed Translator	1
Lexical analysis	1
Finite automata, from regular expression to automata.	2
Syntax analysis.	1
Top-down parsing.	1
Bottom-up parsing.	2
Syntax Directed Translation.	2
Intermediate Code Generation.	1
Problem Solving	1
Project Discussion	1