

Salahadden-Hawler University
College of Science
Computer Department

COURSE BOOK

COMPILER DESIGN

Botcher Degree in science of Computer

Academic year: 2015/2016

Lecturer

Dr. Shaimaa Awad Baha Al_Deen al_aubi

PhD in Computer Science

Third Stage-CS

Office hours: Monday 10:30pm-12:30am, or any other times by appointment.

Class: Monday 8:30am – 10:30am

Lab: Sunday 8:30 am-12:30 pm

Course Objective

Interaction involving humans are most effectively carried out through the medium of language. In computer program a programmer language serves as a mean of communication between the person with a problem and the computer used to help solve it.

This course will cover all principles, methods and examples of compiler system that used in computer and how that system works inside computer in order to make a translation from high level language to low level language (0,1) this language which is only understood by computer machine.

Forms of Teaching:

The course consists of two parts; a theoretical part and practical or applied part, part I (theory) will be depend on lectures in the hall to explain the basic concepts associated with the course by using the Power Point.

Part II (practical) which is associated with training on the use and establishment of databases will be applied in the computer lab.

Grading

The final grade will be based upon the following criteria:

27% Midterm Exams: The students are required to do at least two closed book exam at the mid of semester.

13% Practical Activity: The students are required to perform specified tasks in the lab and provide a small project at the end of course.

60% Final Exam: It will be determined by the exam board.

Course Material

1- Principles of compiler design

Alfred V. Aho & Jeffrey D. Ullman

2- Basics of compiler design

Torben Egidius Mogensen

3- Compilers: principles, techniques, and tools

Alfred V. Aho & Jeffrey D. Ullman

Course Program

Week 1: introduction with definition compiler, translator.

Week 2: the analysis and synthesis of compilation, the structure of compiler.

Week 3: symbol table management, types of errors, error detection and reporting.

Week4: Lexical analysis, token, patterns, lexemes.

Week5: symbol table, input buffer.

Week6: a simple approach to the design of lexical analysis.

Week 7: examples of a simple approach to the design of lexical analysis, specification of tokens

Week8: string and language, quiz.

Week 9: operation on language, regular definitions

Week10: Finite Automata, NFA, examples.

Week11: deterministic finite automata (DFA).

Week 12: syntax analysis, parse tree and derivation, examples.

Week 13: writing grammar, ambiguity, examples

Week14: first mid course exam

Week 15: left recursion, left factoring.

Week 16: top down parser, predictive parsing method

Week 17: predictive parsing algorithm, examples

Week 18: First and Follow

Week 19: First and Follow examples

Week 20: Construction of predictive parsing tables, algorithm of construction of predictive parsing tables

Week 21: examples of construction of predictive parsing tables

Week 22: LL(1) grammars, error detecting and reporting, examples

Week 23: Bottom up parsing, shift reduce parsing method

Week 24: second mid course exam

Week 25: shift reduce parsing method examples

Week 26: operator precedence parser

Week 27: stack implementation of operator precedence parser

Week 28: discussion and general review

Final Exam: will be determined by the exam board.