Zarqa University
Faculty Information Technology
Department: Computer Science
Course title: Compiler Construction (1501421)

Instructor
Lecture's time:
Semester:
Office Hours:

Course description:
This course concerns about how to design and construct compilers of formal programming languages. Topics include compilation goals, organization of a translator, grammar and languages, symbol tables. And presents the formal phases of compilation process: lexical analysis, syntax analysis (parsing), error handling, and intermediate and final code generation.

Aims of the course:

students are expected to:

1. Introducing an overall view of compilation process.
2. Design a context free grammar of a formal language.
3. Implementation of the "analysis phases" of a compiler.
4. Using automated tools to implement syntax analyzers and parsers.
5. Using compiler concepts for other fields such as "pattern recognition".

Intended Learning Outcomes (ILOs):

Upon successful completion of this course, students will be able to:

A. Knowledge and Understanding

A1. Concepts and Theories:

- Understand basic theories used in compilation process
- Understand the concept of compilation
- List the concepts of language theories
- Use tools and techniques to translate a program written by a high level language into machine language.
- Understand the compiler phases
- Build lexical analyzers and use them in the construction of parsers
- Perform the operations of semantic analysis.

A2. Contemporary Trends, Problems and Research:

- Understand recent advances in building compilers especially those that improve the efficiency of compilers.

A3. Professional Responsibility:

- Abide by laws and regulations of software development and design

B. Subject-specific skills

B1. Problem solving skills:

- Analyze and understand the problems affecting the performance of compilers.

B2. Modeling and Design:

- Learn how to design a complier using models and tools.

B3. Application of Methods and Tools:

- Learn how to use technique, methods and tools to translate the program into assembly code.

C. Critical-Thinking Skills

C1. Analytic skills:

- Use analytic skills to analyze complier construction related problems and appropriate solution(s)

C2. Strategic Thinking:

- Use strategic thinking to propose efficient solutions for complex problems

C3. Creative thinking and innovation:

- Use creative thinking and innovation to solve problems affecting negatively the performance of compilers.

D. General and Transferable Skills (other skills relevant to employability and personal development)

Communication: Express and communicate ideas in written and oral forms.

Teamwork and Leadership: Be cooperative members of a team

Organizational and Developmental Skills: Plan, prioritize, and achieve defined goals

Ethical and Social Responsibility: Understand that they are accountable for their actions and there must be a balance between economic growth and the welfare of the society and environment**.**

Course Structure:

Week	Hours	ILOs	Topics	Teaching Procedure	Assessment methods
1	3	A1,A2	**Introduction** • Compilers • Analysis of the Source Program • The Phases of a Compiler • Compiler – Construction Tools	Lecturing with active participation, quizzes, team learning.	Homework, quizzes, reports
2 – 3	6	A1, A2, B1, B2, B3	**Lexical analysis** • The Role of the Lexical Analyzer • Tokens • regular Expression • Specification of Tokens • Recognition Many Tokens	=	=
4 – 7	12	A1, A2, B1, B2, B3, C1	. **Syntax Analysis** • The Role of the Parser • Context – Free Grammars • Writing a Grammar	=	= **FIRST EXAM**

8 – 9	6	A1, A2, B1, B2, B3, C1, C2, C3	**Top-Down parsing and LL Grammar** • Recursive parser • Predictive parsing • Parsing Table • Non recursive Parser	=	=
		A1, A2, B1, B2, B3, C1, C2, C3	**Bottom-Up Parsing and LR Grammar** • Shift-Reduce parser • LR(0) • SLR(1)		
10	3	A1, A2, A3, B1, B2, B3, C1, C2, C3	**Semantic Analysis** • Syntax Directed Translation • Annotated Parse Tree	=	= SECOND EXAM
11 – 12	6	A3, B1, B2, B3, C1, C2, C3, D	**Intermediate code** • Directed Acyclic Graph • Three Address code • Representation trees	=	=
13 – 14	6	A3, B1, B2, B3, C1, C2, C3, D	**Code Generation** • Implementation • Target language output	=	=
15					**FINAL EXAM**

References:

A. Main Textbook:

Aho, "Compilers: Principles, Techniques, and Tools", Addison Wesley, 2nd ed 2007

B. Supplementary Textbook(s):

Jack W. Crenshaw, "Compiler Building Tutorial Let's Build a Compiler" October 10, 2012

Assessment Methods:

Methods	Grade	Date
First Exam	20%	
Second Exam	20%	
Assignments (Reports /Quizzes/ Seminar / Tutorials /Homework….)	10%	
Final Exam	50%	