

Planning and Quality Assurance Affairs

Form (A)

Course Specifications

General Information

Course name	Compilers Design
Course number	ITCS4317
Faculty	
Department	
Course type	Major Needs
Course level	4
Credit hours (theoretical)	3
Credit hours (practical)	0
Course Prerequisites	

Course Objectives

- 1 - To recognize and use terminology and formalisms related to grammars for programming languages and compiler construction
- 2 - To understand the elementary of intermediate code representation, symbol tables, run- time structures, code generation and compiler construction tools
- 3 - To identify and describe the concepts underlying the components of a compiler including a scanner, parser, and code generator
- 4 - Constructing the basic components of a compiler including the scanner, parser, code generator, and symbol table

Intended Learning Outcomes

Knowledge and Understanding	<ul style="list-style-type: none">* a1) Explain the compiler parsing techniques, construction of abstract syntax trees, and symbol tables* a2) Describe the intermediate machine representations and actual code generation* a3) Identify the theory and practice of compiler implementation* a4) Identify the core of analysis and statistics in the compiler theory* a5) Provide a deeper understanding of some aspects of compiler design
Intellectual Skills	<ul style="list-style-type: none">* b1) Expressing languages formally using regular expressions and context-free grammars* b2) Perform type analysis of programs using the syntax-directed translation technique* b3) Show how activation records are used when executing programs containing more than one function* b4) Perform comparisons between (algorithms, methods, techniques...etc* b5) Identify attributes, components, relationships, patterns, main ideas, and errors* b6) Apply the concepts, principles, theories and practices underpinning computing as an academic discipline
Professional Skills	<ul style="list-style-type: none">* c1) Write parsers using a formal description of the language* c2) Use tools for generating compilers* c3) Specify, design, and implement compilers using appropriate programming language* c4) Deploy effectively the tools used for documentation of software
General Skill	<ul style="list-style-type: none">* d1) Communicate effectively by oral, written and visual means* d2) Work effectively as an individual and as a member of a team* d3) Lead and motivate individuals. d4. Manage tasks and resources* d4) Manage tasks and resources

Course Contents

<ol style="list-style-type: none">1 - Structure of compiler, lexical analysis, lexical patterns, deterministic & Nondeterministic finite automata, scanner, construction, limits of regular languages, derivations, parse trees, ambiguous grammars, Top-down parsers algorithms, Bottom-up parsers algorithms, table construction, hierarchy of context-free languages, context sensitive analysis, procedure abstraction2 - Introduction to code generation, code shape and arithmetic expressions3 - Code optimization and static analysis, data-flow analysis, data-dependence analysis, transformations, taxonomy, scalar transformations, post-pass optimizations4 - Instruction selection instruction, scheduling, register allocation
--

Teaching and Learning Methods

<ol style="list-style-type: none">1 - Lectures2 - Practical Exercises3 - Projects4 - Case Study
--

Teaching and Learning Methods for the Disabled Students

<ol style="list-style-type: none">1 - ---

Students Assessment

<u>Assessment Method</u>	<u>TIME</u>	<u>MARKS</u>
Mid-Term Exam I / Project I	6th week	20
Mid-Term Exam II / Project II	12th week	20
Class Work	During the 16 week	10
Final exam	16th week	50

Books and References

Course note	Lecture Course Notes
Essential books	Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, "Compilers: Principles, Techniques, & Tools", Addison Wesley, 2 edition, 2007.
Recommended books	John Levine, "Flex & Bison: Text Processing Tools", O'Reilly Media, 1 edition, 2009

Knowledge and Skills Matrix

Main Course Contents	Study Week	Knowledge and Understanding	Intellectual Skills	Professional Skills	General Skill
Structure of compiler, lexical analysis, lexical patterns, deterministic & Nondeterministic finite automata, scanner, , construction, limits of regular languages, procedure abstraction	1-5	a1, a3, a5	b1, b2, b4, b5, b6	c1, c2, c3, c4	d1, d2, d3, d4
derivations, parse trees, ambiguous grammars, Top-down parsers algorithms, Bottom-up parsers algorithms, table construction, hierarchy of context-free languages, context sensitive analysis	6-8	a1, a3, a5	b1, b2, b4, b5, b6	c1, c2, c3, c4	d1, d2, d3, d4
Introduction to code generation, code shape and arithmetic expressions.	9-10	a2, a3, a4	b4,b5, b6	c2, c3	d1, d2, d3, d4
Code optimization and static analysis, data-flow analysis, data-dependence analysis, transformations, taxonomy, scalar transformations, post-pass optimizations.	11-13	a2, a3, a4	b3, b4, b5, b6	c2, c3	d1, d2, d3, d4
Instruction selection instruction scheduling, register allocation	14-15	a3	b4-b6	c3	d1, d2, d3, d4