Course Specifications

Course Title:	Compiler Design
Course Code:	ICS 413
Program:	**Computer Science**
Department:	**Computer Science and Information**
College:	**Science Az Al-Zulfi**
Institution:	**Majmaah University**

Table of Contents

A. Course Identification

1. Credit hours:3
2. Course type **a.** University ☐ College ☐ Department ☑ Others ☐ **b.** Required ☐ Elective ☑
3. Level/year at which this course is offered 7th Level
4. Pre-requisites for this course (if any)**: Finite Automata and Computability ICS 321** **Object Oriented Programming ICS211**
5. Co-requisites for this course (if any)**:Nil**

6. Mode of Instruction (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	**Traditional classroom**	48	80%
2	**Blended**	6	10%
3	**E-learning**	6	10%
4	**Correspondence**	-	-
5	**Other**	-	-

7. Actual Learning Hours (based on academic semester)

No	Activity	Learning Hours
Contact Hours		
1	**Lecture**	30
2	**Laboratory/Studio**	30
3	**Tutorial**	
4	**Others** (specify)	
	Total	60
Other Learning Hours*		
1	**Study**	30
2	**Assignments**	30
3	**Library**	
4	**Projects/Research Essays/Theses**	10
5	**Others** (specify)	30
	Total	100

***** The length of time that a learner takes to complete learning activities that lead to achievement of course learning outcomes, such as study time, homework assignments, projects, preparing presentations, library times

B. Course Objectives and Learning Outcomes

1. Course Description

Compiler Design course introduces the design and implementation of compilers. Topics include:

1. compiler organization,
2. algorithms for lexical,
3. syntactic and semantic analysis,
4. top-down and bottom-up parsing
5. symbol table organization,
6. error detection and recovery,
7. intermediate and object code generation,
8. code optimization.

2. Course Main Objective

1. To describe structures (i.e., "formulas") using grammars;
2. To parse, i.e., to recognize (build) such structures in (from) a sequence of symbols;
3. To analyze grammars to see whether or not specific properties hold;
4. To compose components such as parsers, analyzers, and code generators
5. To apply these techniques in the construction of all kinds of programs;
6. To familiarize oneself with the concept of computability.

3. Course Learning Outcomes

	CLOs	Aligned PLOs
1	**Knowledge:**	
1.1	Understand the structure of compilers	**ICS-a2**
1.2	Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation.	**ICS-a2**
1.3	Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines.	**ICS-a2**
1.4	Explain the core issues of Compiler design.	**ICS-a2**
2	**Skills :**	
2.1	Design and implement a compiler using a software engineering approach be able to conduct HCI evaluations and usability studies.	b1
2.2	Identify problems, and explain, analyze, and evaluate various design strategies of compilers.	b1
3	**Competence:**	
3.1	Use Information technology and computer skills to gather information about a selected topic.	c3

C. Course Content

No	List of Topics	Contact Hours
1	**Introduction to Compilers**: The role of language translation in the programming process, Comparison of interpreters and compilers, Language translation phases, Machine dependent and machine independent aspects of translation, Language translation as a software engineering activity.	4
2	**Lexical Analysis**: Application of regular expressions in Lexical Analysis, Scanning, hand coded scanner vs. automatically generated scanners, formal definition of tokens, implementation of finite state automata	12
3	**Syntax Analysis and Parsing**: Revision of formal definition of grammars, BNF and EBNF, Bottom-up vs. Top-down parsing, Tabular vs. Recursive-descent parsers, Error handling.	12
4	**Parser Generators**: Automatic generation of tabular parsers, Symbol table management, Use of tools in support of the translation process.	8
5	**Semantic Analysis**: Data type as set of values with set of operations, data types, Type- checking models, Semantic models of User defined types, Parametric polymorphism, Subtype polymorphism, Type checking algorithms.	8
6	**Intermediate Code Generation:** Intermediate and object code, intermediate representations, implementation of code generators**.**	12
Total		56

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge**		
1.1	Understand the structure of compilers	Lectures Lab demonstrations Case studies Individual presentations	Written Exam Homework assignments Lab assignments Class Activities Quizzes
1.2	Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation.		
1.3	Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack		

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
	machines.		
1.4	Explain the core issues of Compiler design.		
2.0	**Skills**		
2.1	Design and implement a compiler using a software engineering approach be able to conduct HCI evaluations and usability studies.	Lectures Lab demonstrations Case studies Individual presentations Brainstorming	Written Exam Homework assignments Lab assignments Class Activities Quizzes Observations
2.2	Identify problems, and explain, analyze, and evaluate various design strategies of compilers.		
3.0	**Competence**		
3.1	Use Information technology and computer skills to gather information about a selected topic.	Small group discussion Whole group discussion Brainstorming Presentation	Written Exam Homework assignments Lab assignments Class Activities Quizzes

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	First written mid-term exam	6	15%
2	Second written mid-term exam	12	15%
3	Presentation, class activities, and group discussion	Every week	10%
4	Homework assignments	After each chapter	10%
5	Implementation of presented protocols	Every two weeks	10%
6	Final written exam	16	40%
7	Total		100%

***Assessment task** (i.e., written test, oral test, oral presentation, group project, essay, etc.)

E. Student Academic Counseling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice :

Office hours: Sun: 10-12, Mon. 10-12, Wed. 10-12

Email: m.wagieh@mu.edu.sa

F. Learning Resources and Facilities

1.Learning Resources

Required Textbooks	Affred V. Aho, Monica S. Lam, and Ravi Sethi, Compilers: Principles, Techniques & Tools, edition 2nd , Addision Wesley, 2014, 978-0321486813
Essential References Materials	Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J.H. Jacobs, Koen Langendoen, Modern Compiler Design, edition 2nd, Springer, 2012, 978-1461446989.
Electronic Materials	-
Other Learning Materials	-

2. Facilities Required

Item	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Classroom and Labs as that available at college of science at AzZulfi are enough.
Technology Resources (AV, data show, Smart Board, software, etc.)	Smart Board
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	N/A

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Questionnaires (course evaluation) achieved by the students and it is electronically organized by the university.	Students	Indirect
Student-faculty management meetings.	Program Leaders	Direct
Discussion within the staff members teaching the course	Peer Reviewer	Direct
Departmental internal review of the course.	Peer Reviewer	Direct
Reviewing the final exam questions and a sample of the answers of the students by others.	Peer Reviewer	Direct
Visiting the other institutions that introduce the same course one time per semester.	Faculty	Indirect

Evaluation areas (e.g., Effectiveness of teaching and assessment, Extent of achievement of course learning outcomes, Quality of learning resources, etc.)
Evaluators (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)
Assessment Methods (Direct, Indirect)

H. Specification Approval Data

Council / Committee	Dr. Mohamed Wagieh
Reference No.	
Date	08/09/2019