هيئة تقويم التعليم والتدريب
Education & Training Evaluation Commission

T-104
2022

Course Specification

Course Title:	**Theory of compilers**
Course Code:	**CS1511**
Program:	**Computer Science**
Department:	**Computer Science and Engineering**
College:	**Computer Science and information technology**
Institution:	**Albaha University**
Version: :	**T104 – V1**
Last Revision Date:	February 9, 2023

Table of Contents:

A. General information about the course:

Course Identification					
1. Credit hours:	3				
2. Course type					
a.	University ☐	College ☐	Department☒	Track☐	Others☐
b.	Required ☒	Elective☐			
3. Level/year at which this course is offered: 11 / 4th year					

4. Course general Description

Lecture:

This is a course in the theory and design of compilers using modern concepts. Students learn the basic elements of a language translator (compiler); lexical analysis, parsing, code generation, symbol table management, type checking, scope resolution, code optimization, and error recovery. They also learn to write regular expressions and context free grammars and understand the separate phases of compilation and the issues involved in designing a medium sized translator. To facilitate student understanding, a semester-long, incremental design project is employed. As a result of building their own compiler, students learn the operation and messages presented by any modern commercial translator.

Lab:

The lab is designed to give students practical experiments on compiler and familiarity with compiled codes (assembly language). Students will learn how to:

- Write a scanner and a predictive parser for a small language,
- Perform a small experiment with scanner (lex/flex) and parser generator (such as translation of regular expressions to NFA or the construction of parse tree),
- Write a scanner parse specification for a small language, translation of the language to an intermediate form (e.g. three-address code), and generation of target code (in assembly language).

5. Pre-requirements for this course (if any): Theory of Computation (CS1507)

6. Co- requirements for this course (if any): none

7. Course Main Objective(s)

The main purpose for this course is to teach students how to:

- Describe the component of a compiler works.
- List of algorithms used in building compilers and their connections to system hardware
- Describe parsing theory and grammar implementation.
- Implement a simple working compiler.
- Explain the structure of compilers using a novice software compiler
- Develop compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation

- Demonstrate the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines
- Interact in groups collaboratively.
- Communicate concepts and techniques in oral presentations.

1. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1.	Traditional classroom	44	100%
2.	E-learning		
3.	Hybrid • Traditional classroom • E-learning		
4.	Distance learning		

2. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	22
2.	Laboratory/Studio	22
3.	Field	-
4.	Tutorial	-
5.	Others (specify)	-
	Total	44

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Describe the component of a compiler works.	K1	- Lectures - Slides presentation - Multimedia presentation	- Home work (rubric) - Quizzes - Midterm exam - Final Exam
1.2	Describe parsing theory and grammar implementation	K2	- Lectures - Slides presentation - Multimedia presentation - Discussions	- Home work (rubric) - Quizzes - Midterm exam - Final Exam
2.0	Skills			
2.1	Implement a simple working compiler.	S1	- Demonstrations - Labs Lectures - Group Discussion - Group Projects - Case Studies	- Home work (rubric) - Quizzes - Final Exam - Project evaluation form (rubric)
2.2	Explain the structure of compilers using a novice software compiler	S2	- Demonstrations - Labs Lectures - Group Discussion - Group Projects - Case Studies - Practical Exercises	- Homework/Assignments - Quizzes - Midterm Exam - Final Exam - Lab exercises (Rubric) - Lab exams - Viva-voce (Rubric) - Project Assessment (Rubric) - Report Assessment (Rubric)
2.3	Develop compiler construction such as lexical analysis, top-down, bottom-up parsing,	S2	- Demonstrations - Labs Lectures - Group Discussion	- Homework/Assignments - Quizzes

1.	Introduction to Compiling	2
2.	Lexical Analysis.	3
3.	Syntax Analysis	3
4.	Syntax Directed Analysis	3
5.	Type Checking	2
6.	Run time Environments	2
7.	Intermediate Code Generation.	1
8.	Code Generation.	3
9.	Code Optimizations	3
Total		22

No	List of Topics (Lab)	Contact Hours
1.	Lexical Analysis.	4
2.	Syntax Analysis	4
3.	Syntax Directed Analysis	4
4.	Type Checking	3
5.	Intermediate Code Generation.	3
6.	Code generation	4
Total		22

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Homework	Every two Weeks	10%
2.	Midterm	6	20%
3.	Quiz	10	10%
4.	Lab activities and Lab Exam	12	20%
5.	Final Exam	13	40%
6.	Lab reports	Every four Weeks	5%
	Total		**100%**

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.)

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools (2nd Edition), Addison Wesley, Boston, MA, 2006. ISBN 0321486811

Supportive References	- Computer Science Curriculum 2013 – http://cs2013.org - ACM (Association for Computer Machinery) Curricula Recommendations - • http://www.acm.org/education/curricula-recommendations
Electronic Materials	• ACM (Association for Computer Machinery) web site - http://www.acm.org/ • IEEE Computer Society web site - http://www.computer.org/portal/web/guest/home • Access to the Saudi Digital Library (SDL). • Using the learning management system of the university – Rafid System (https://lms.bu.edu.sa/).
Other Learning Materials	C++ Using Visual Studio as compiler

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	• A classroom or lecture hall with whiteboard for 25 students. • A digital circuit's laboratory.
Technology equipment (projector, smart board, software)	• A digital image projection system with connection to desktop computer and laptop computer. • High speed Internet connection. • An instructor computer station.
Other equipment (depending on the nature of the specialty)	None

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	• Students • Faculty • Peer Reviewers • Program Leader • Course Coordinator	• Surveys (indirect). • Direct feedback from students (interview between Program leader and students). • Course evaluation by Peer Reviewers (indirect). • Class visit by Program Leader • Comprehensive Course report (where we can find information about teaching difficulties and action plan, …)

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of students assessment	• Students • Faculty • Peer Reviewers • Course Coordinator • Exam Evaluation Committee • Course Coordinator	• Surveys (indirect). • Direct feedback from students (interview between Program leader and students). • Assessment results (direct) • Course evaluation by Peer Reviewers (indirect). • Comprehensive Course report (where we can find information about assessment difficulties and action plan, …) • Exam evaluation by the Exam Evaluation Committee (indirect)
Quality of learning resources	• Students • Faculty • Peer Reviewers • Course Coordinator	• Surveys (indirect) • Course evaluation by Peer Reviewers (indirect). • Comprehensive Course report (where we can find information about difficulties and challenges about learning resources as well as consequences and action plan, …)
The extent to which CLOs have been achieved	• Faculty • Program Leader • Course Coordinator •	• Student Results (direct) • Comprehensive Course report (where we can find the CLO assessment results)
Other	None	None

Assessor (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify)
Assessment Methods (Direct, Indirect)

G. Specification Approval Data

COUNCIL /COMMITTEE	
REFERENCE NO.	
DATE	