Academic Program Description Form

University Name: Tikrit University

Faculty/Institute: .....

Scientific Department: .....

Academic or Professional Program Name: .....

Final Certificate Name: .....

Academic System: .....

Description Preparation Date:

File Completion Date:

Signature:

Head of Department Name:

Date:

Signature:

Scientific Associate Name:

Date:

The file is checked by:

Department of Quality Assurance and University Performance

Director of the Quality Assurance and University Performance Department:

Date:

Signature:

Approval of the Dean

1. Program Vision
Program vision is written here as stated in the university's catalogue and website.
2. Program Mission
Program mission is written here as stated in the university's catalogue and website.
3. Program Objectives
General statements describing what the program or institution intends to achieve.
4. Program Accreditation
Does the program have program accreditation? And from which agency?
5. Other external influences
Is there a sponsor for the program?

6 Program Structure				
Program Structure	Number of Courses	Credit hours	Percentage	Reviews•
Institution Requirements				
College Requirements				

Department Requirements	3	3	%100	basic
Summer Training				
Other				

This can include notes whether the course is basic or optional.

7. Program Description

Year/Level	Course Code	Course Name	Credit Hours	
			theoretical	practical
2023/2024		Compiler2	30	30

8. Expected learning outcomes of the program

Knowledge	
Learning Outcomes 1	Learning Outcomes Statement 1
Skills	
Learning Outcomes 2	Learning Outcomes Statement 2
Learning Outcomes 3	Learning Outcomes Statement 3
Ethics	
Learning Outcomes 4	Learning Outcomes Statement 4
Learning Outcomes 5	Learning Outcomes Statement 5

9. Teaching and Learning Strategies

Teaching and learning strategies and methods adopted in the implementation of the program in general.

10. Evaluation methods

Implemented at all stages of the program in general.

11. Faculty

Faculty Members

Academic Rank	Specialization		Special Requirements/Skills (if applicable)		Number of the teaching staff	
	General	Special			Staff	**Lecturer**

Professional Development

Mentoring new faculty members

Briefly describes the process used to mentor new, visiting, full—time, and part—time faculty at the institution and department level.

Professional development of faculty members

Briefly describe the academic and professional development plan and arrangements for faculty such as teaching and learning strategies, assessment of learning outcomes, professional development, etc.

12. Acceptance Criterion

(Setting regulations related to enrollment in the college or institute, whether central admission or others)

13. The most important sources of information about the program

State briefly the sources of information about the program.

14. Program Development Plan

Program Skills Outline															
				Required program Learning outcomes											
Year/Level	Course Code	Course Name	Basic or optional	Knowledge				Skills				Ethics			
				A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
2023 – 2024		Compiler2	Basic	*	*	*	*	*	*	*	*	*	*	*	*

- **Please tick the boxes corresponding to the individual program learning outcomes under evaluation.**

Course Description Form

1. Course Name: Compiler2
2. Course Code:
3. Semester / Year: 2023 – 2024 First Semester
4. Description Preparation Date: **2024 / 9 / 10**
5. Available Attendance Forms: In attendance lectures
6. Number of Credit Hours (Total) / Number of Units (Total) : 60 / 3
7. Course administrator's name (mention all, if more than one name)
Name: Lecturer. Mohanad Dawood Salman Email: mohanaddawoodalroomi@tu.edu.iq Name: **Assistant Lecturer: Luay Ibrahim lalif** Email: luay.i.khalaf@tu.edu.iq .....
8. Course Objectives

Course Objectives	• **Student be able to Identify principles, techniques and tools for compilers of programming languages.** • **Student be able to design a compiler for a (simplified) (programming) language.** • **Student know how to use compiler construction tools, such as generators of scanners and parsing.** • **Student be familiar with assembly code and virtual machines.** • **Student be familiar with compiler analysis and optimization techniques.**
9. Teaching and Learning Strategies	
Strategy	• **Using active learning: This type of learning includes engaging learners in active and interactive learning processes, such as discussions, practical experiments, educational games, and knowing the students' scientific level by evaluating their test results.** • **Technology-based learning: includes the use of technology in learning and teaching processes, such as the use of multimedia, educational applications, and**

	online educational platforms. Technology can help improve access to knowledge and enhance interaction and engagement.

10. Course Structure

Week	Hours	Required Learning Outcomes	Unit or subject name	Learning method	Evaluation method
1.	**2 hours theoretical**	**Recognition is the fourth stage in the programming language compiler.**	**Intermediate code Generation.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** • **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write an intermediate code generation program in C++.**	**Program of intermediate code Generation in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** • **H.W: assignments through electronic classroom.**
2.	**2 hours theoretical**	**Gain knowledge about writing triple address code.**	**Three-address code**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write a three-address code program in C++.**	**Program of Three-address code in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
3.	**2 hours theoretical**	**Knowledge of types of advertisements and translation of arithmetic expressions.**	**Types and declarations, Translation of expressions.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write a program to structure constraint matrices.**	**Structures (struct_array) in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
4.	**2 hours theoretical**	**Identify graphical types and control traceability of**	**Type Checking, Control Flow.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic**

		programming commands.			**classroom.**
	2 hours Lap	**Learn how to write a vocabulary table program and create a symbol table in C++.**	**Lexemes table program, Token table program in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
5.	**2 hours theoretical**	**Know the term Backpatching. And install the Switch-Statements statement..**	**Backpatching. Switch-Statements.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write an identifier table program in C++.**	**Symbol table program in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
6.	**2 hours theoretical**	**Knowledge of the intermediate code structure of subprograms.**	**Intermediate Code for Procedures.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write an increment and decrement program in C++.**	**Increasing and decreasing program in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
7.	**2 hours theoretical**	**Knowledge of the fifth phase of the compiler, the code optimization phase for programming languages.**	**Code Optimization. Examples of code Optimization.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write the #1 code optimization program in C++.**	**Code Optimization program1 in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
8.	**2 hours theoretical**		**Theoretical exam(1)**		
	2 hours Lap		**Practical exam(1)**		

9.	**2 hours theoretical**	**Knowing the sixth phase of the compiler, the code generation phase, as well as the target code.**	**Gode generation. The target language.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write code optimization program No. 2 in C++.**	**Code Optimization program2 in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
10.	**2 hours theoretical**	**Know the classification of memory addresses in the target code.**	**Addresses in the Target code.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write a transition tracking program in C++.**	**Control Flow program in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
11.	**2 hours theoretical**	**Know the basic rules of blocks and transitions.**	**Basic Blocks and flow Graphs.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write an iteration program and represent it in three-address code in C++.**	**three address code program of loop in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
12.	**2 hours theoretical**	**Know the simple basics about generating object code.**	**A Simple Code Generator.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write a program to reserve memory registers in C++.**	**Registration booking program in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
13.	**2 hours theoretical**	**Learn how to record the locations and substitutions of the target code.**	**Register Allocation and Assignment**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**

	2 hours Lap	**Learn how to write the final code generation program No. 1 in C++.**	**Code generation program1 in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
14.	**2 hours theoretical**	**Knowing machine language and adopting its improvement.**	**Machine – Independent Optimizations.**	• **Presentation.** • **Discussion.** • **Brainstorming.**	• **Quiz.** **H.W: assignments through electronic classroom.**
	2 hours Lap	**Learn how to write the final code generation program No. 2 in C++.**	**Code generation program2 in c++ language.**	• **Presentation.** • **Discussion.**	• **Testing students through practical performance.** **H.W: assignments through electronic classroom.**
15.	**2 hours theoretical**		**Theoretical exam(2)**		
	2 hours Lap		**Practical exam(2)**		

11. Course Evaluation

Theoretical exam(1) **10%**	**Practical exam(1)** **5%**	**Theoretical exam(2)** **10%**	**Practical exam(2)** **5%**	H.W. and Quiz **10%**	Final exam 60%	The final grade%100
10	**5**	**10**	**5**	**10**	**60**	**100**

12. Learning and teaching resources

Required textbooks	
Main references (sources)	**Compilers: Principles, Techniques, and Tools" by Aho, Sethi, and Ullman, 2nd edition. (2006).**
Recommended books and references (scientific journal and reports)	1. **Waite, W. M., & Goos, G. (2012). Compiler construction. Springer Science & Business Media.** 2. **Mogensen, T. Æ. (2009). Basics of compiler design. Torben Ægidius Mogensen.** **من البداية إلى البرمجة الكيانية ، الدكتور المهندس. نضال خضير العبادي ، C++ .(2011)**
Electronic references, websites	https://www.youtube.com/watch?v=SMkQcn1ihLw&list=PL9fwy3NUQKwZe1P-Tr_n9TWAgawgGkpm5