

Ministry of Higher Education and Scientific Research

Supervision and Scientific Evaluation Body

Quality Assurance and Academic Accreditation Office

## Course Description Sample

**Subject:** -----Compiler-----

This course description provides a brief survey of the most important characteristics, expected learning output, showing whether students have made full use of the learning opportunities. These characteristics have to be matched with the description of the program.

1. Educational Institution	Shatt Al-Arab University College
2. Department / Center	Computer Science
3. Course Title /Code	Compiler
4. Lecturer Name	Abdulhussein latef Khudheir
5. Type of Teaching	Attendance
6. Academic Year /Term	2022-2023
7. Total No. of Teaching Hours	4
8. Date of Preparing this Course Description	28-9-2022

### 9. Course Objectives

**The course aims to study and know the compiler of higher-level programming languages and how to convert those higher-level languages into machine language.**

**The student studies the stages of program implementation and the phases that the program goes through, up to the stage of implementation on the computer**


## 10. Course Output, Methodology and Evaluation

### (A) Cognitive Objectives

<b>a. That the student be able to know and understand the subject of translations.</b>
<b>b. Understand the steps necessary to convert any program from the source language to the language of the machine.</b>
<b>c. That the student recognizes the most important phases that the program goes through during the translation process.</b>
<b>d. Identifying the mechanisms for identifying program errors and how to correct them.</b>
<b>e. Distinguish between interpreted languages and composed languages.</b>

### (B) Skill Objectives Related to the Program:

<b>a. Enabling the student to practically simulate the programming of the compiler phases.</b>
<b>b. Enable the student to understand the errors generated when implementing programs and how to correct them.</b>
<b>c. Increase the student's ability to apply the algorithms for the work of the translator.</b>

### Methods of Teaching and Learning

<b>The traditional theoretical learning method in the classroom with the use of the electronic presentation method.</b>
<b>Practical learning method in computer laboratories.</b>
<b>The method of communication and e-learning through the Google Classroom learning platforms.</b>

### Methods of Evaluation

<b>a. Activating student activity by giving daily assignments orally and in writing.</b>
<b>b. Conducting quick monthly exams during lectures to activate the student's role in constantly reviewing the material.</b>
<b>c. Continuous student communication in lectures and not absenteeism.</b>
<b>d. Central semester exams for the two semesters.</b>

### (C) Sentimental and Value Objectives

<b>a. Positive interaction between the student and the professor by simplifying the learning process and bringing the material to the student's awareness.</b>
<b>b. Spreading the spirit of cooperation and love in the classroom between the student, his colleagues and the professor of the subject to create a distinguished scientific atmosphere.</b>
<b>c. Opening the doors of discussion and exchanging questions and knowledge in the classroom and moving away from the traditional method of restricting the student to discussion or questioning and removing embarrassment in the participation of students in the topic of the lecture.</b>
<b>d. Activating intellectual and creative activity in developing plans to reach solutions in different ways.</b>

### Methods of Teaching and Learning

a. Lectures on university instructions.
b. Educational guidance lectures.
c. Continuous directing.
d. Visiting State and private institutions.
e. Showing practical cases.

### Methods of Evaluation

a. Daily quizzes.
b. Classroom discussions and commitment to ethics and sublime values.
c. Special marks for class activities.
d. Monthly and quarterly evaluation.

### D) General and Qualitative Skills (other skills related to the ability of employment and personal development)

<b>a. The student's ability to teach the subject .</b>
<b>b. The possibility of simulating the author's work programmatically.</b>
<b>c. Enable the student to understand programs and design algorithms.</b>
<b>d. The ability to solve all errors in the program.</b>

## 11. Course Structure

Week	No of Hours	Required Learning Output	Title of Subject	Teaching Method	Evaluation
1	4	Knowing the concept of the compiler	Definition of the compiler	lecture + lab	Daily and monthly exam, activities and attendance
2	=	Find the difference between interpreted languages and compiler languages	Interpreter and compiler and the difference between them	=	=
3		Learn about different types of compilers	Types of compiler and ways to deal with programs		
4		Knowing the structure of the compiler's program	Compiler program structure		
5		Get to know the compiler phases	An overview of the compiler phases		
6		Understand the outputs of the two phases	Read and lexical analysis phase		
7		Knowing the outputs of the syntax phase	Syntax analysis phase		
8		Knowing the specification of the names in the program and the coding of the tree nodes	Descriptor and simple translation phase		
9		Knowledge of compiler optimization tools	Loading and Optimization		
10		Definition of the compiler's other work	Extra job of compiler		
11		Entering the details of the phases	The details of compiler phase		
12		Know the ASCII code series	The characters in Read phase		
13		Know how to represent the paragraphs in the	Represent of the items and item's table		

		<b>table</b>			
14		<b>Knowing the composition of the symbol table, its importance and how to access the symbols</b>	<b>Symbol table and search ways</b>		
15		<b>Understand and know the formation of the parsing tree and the connection between paragraphs</b>	<b>Pars Tree and triples</b>		
16		<b>Understand the work of the analyzer with arithmetic expressions</b>	<b>Bottom-Up parsing</b>		
17	=	<b>Understand the work of the analyst with instructions</b>	<b>Top-Down parsing</b>		
18		<b>Correct the error and handling</b>	<b>Syntax Error handling</b>		
19		<b>Know the problem of the backtracking approach and how to overcome it</b>	<b>Backtracking problem</b>		
20		<b>Understand the Recursive problem of and how to get rid of it</b>	<b>Recursive from left problem</b>		
21		<b>Knowing the existence of interference indirectly and how to detect it</b>	<b>One symbol look head analysis</b>		
22		<b>Define one track and end the backtracking approach</b>	<b>One track analysis</b>		
23		<b>Defining the Ambiguity of the Null symbol</b>	<b>Ambiguity in context</b>		
24		<b>Define the precedence of the terminal symbols</b>	<b>Operator precede</b>		
25		<b>Matrix design to prioritize terminal symbols</b>	<b>Precede matrix</b>		

26		Define a table of symbols with full specifications for names.	Descriptor of symbol table		
27		Understanding cipher output for instructions	Machin language and simple translation		
28		Define how to switch avatar addresses	The loader and object program		
29		Examples of the target program	Exempl-1 :Codes production		
30		Examples of the goal program	Example-2:'=' operation code		

## 12. Infrastructure

a. Textbooks	<b>Compilers: Principles, Techniques, and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman</b>
b. References	<b>Introduction to Compilers and Language Design: Second Edition, author: "Douglas Thain"</b>
c. Recommended books and periodicals (journals, reports, etc.)	<b>1-introduction to compiler design by torben, and Egidius mogensen. 2-compiler construction for digital computers, by David Gries</b>
d. Electronic references, internet websites, etc	

## 13. The Plan of Improving the Course

<b>1-Adding Token, pattern and lexemes to lexical analyser</b>
<b>2- Adding Attributes for token to lexical analyser</b>
<b>3- Adding Ambiguity to Syntax analyser</b>