Menoufiya University
Faculty of Computers and Information
Department of Computer Science

جامعة المنوفية
كلية الحاسبات والمعلومات
قسم علوم الحاسب

COURSE SPECIFICATION

(COMPILER DESIGN)

Programme(s) on which the course is given	Computer Science
Major or Minor element of programs	Major
Department offering the program	Computer Science
Department offering the course	Computer Science
Academic year / Level	4th Year / 2nd Semester

A- Basic Information

Title	**Compiler Design**			**Code**	**CS435**	
Credit Hours	**Lecture**	**3**	**Tutorial**	**3**	**Practical**	**-**
	Total				**6**	

B- Professional Information

1- Overall Aims of Course

- To be aware of the distinctions among language translation systems (Compilers, interpreters).
- To understand how language design and implementation are related
- To gain experience with formal language-theoretic techniques
- To understand how storage is managed during the execution of a program.

2- Intended Learning Outcomes of Course (ILOs)

2a- Knowledge and understanding

a2 Understand and apply a wide range of principles and tools available to the software engineer, such as design methodologies, choice of algorithm, language, software libraries and user interface technique.

a7 Understand The basics of the software life cycle, from requirements definition to development and evaluation.

2b- Intellectual skills

b1 Solve a wide range of problems related to the analysis, design and construction of computer systems

b2 Analyze the requirements of a range of computer-based systems and examine the design alternatives based on the constraints imposed by society, organizations, and technology.

2c- Professional and practical skills

c4 Use the scientific literature effectively and make discriminating use of Web resources.

c5 Design, write and debug computer programs in appropriate languages.

c6 Use appropriate computer-based design support tools

2d- General and transferable skills

d1 Display an integrated approach to the deployment of communication skills .

d2 Use IT skills and display mature computer literacy.

d6 Employ discrete mathematical skills as appropriate.

d9 Choose and formulate suitable strategies to accomplish well-defined goals.

3- Contents

Topic	No. of Hours	Lecture	Tutorial /Practical
1 Introduction	6	3	3
2 Lexical Analysis • Formal Languages. • Implementation with Finite State Machines. • Lexical Tables	6	3	3
3 Syntax Analysis • Grammars, Languages, and Pushdown Machines. • Ambiguities in Programming Languages. • The Parsing Problem.	12	6	6
4 Top Down Parsing • Relations and Closure. • Simple Grammars. • Quasi-Simple Grammars. • LL(1) Grammars. • Parsing Arithmetic Expressions Top Down. • Syntax-Directed Translation. • Attributed Grammars. • An Attributed Translation Grammar for Expressions.	12	6	6
5 Bottom Up Parsing • Shift Reduce Parsing. • LR Parsing With Tables	12	6	6

6 Code Generation • Introduction to Code Generation. • Converting Atoms to Instruction. • Single Pass vs. Multiple Passes. • Register Allocation.	12	6	6
7 Optimization • Introduction and View of Optimization. • Global Optimization. • Local Optimization.	12	6	6
8 Implementation Projects in Compiler Design	12	6	6
Total number of Hours for the course	**84**	**42**	**42**

4- Teaching and Learning Methods1

4.1- Lectures
4.2- Exercises and tutorials
4.3- Research assignments

5- Student Assessment Methods

5-a Methods

5.a1- Reports, assignments, exercises, and final written exam to assess knowledge and understanding.
5.a2- Regular oral , and written quizzes to assess intellectual skills.
5.a3- Practical projects, final practical and oral exams to assess professional skills.
5.a4- Reports, assignments, and discussions to assess general and transferable skills

5-b Assessment Schedule

Assessment 1	7^{th} week
Assessment 2	16^{th} week *(Oral).*
Assessment 3	17^{th} -18^{th} weeks *(final written exam)*

5-c Weighting of Assessments

Reports, practical projects, assignments, punctuality and individual class	10%

activity	
Mid-Term Examination	10%
Final oral exams	10%
Final written exam	70%
Total	100%

6- List of References

6-a Course Notes

"Lectures in Compiler Design ", selected by A. Elsisi, 2nd Semester 2006.

6-b Essential Books (Text Books)

[1] Andrew W. Appel Modern Compiler Implementation in Java, Cambridge University Press, 1998.

6-c Recommended Books

[1] Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, Compilers: Principles, Techniques, and Tools Addison-Wesley, 1986.

6-d Periodicals, Web Sites, … etc

IEEE transactions on computers, software

7- Facilities Required for Teaching and Learning

- PC laboratory. .
- Datashow, screen, and laptop computer

Course coordinator:

Dr. Ashraf Elsisi

Head of Department:

Prof. Nabil Abd El-Wahed Ismail

Date: / /