



NATIONAL UNIVERSITY OF ENGINEERING
COLLEGE OF SCIENCES
COMPUTER SCIENCE PROGRAM

CC461 – COMPILERS

I. GENERAL INFORMATION

CODE	: CC461 – Compilers
SEMESTER	: 7
CREDITS	: 4
HOURS PER WEEK	: 6 (Theory – Laboratory)
PREREQUISITES	: CC302 Distributed parallel language oriented to objects CC342 Theory of computing
CONDITION	: Mandatory

II. COURSE DESCRIPTION

That the student knows and understands the concepts and fundamental principles of compilation theory to perform the construction of a compiler. Know the basic techniques used during the process of intermediate generation, optimization and code generation. Learn to implement small compilers.

III. LEARNING UNITS

1. Overview of the Programming Languages

- I: History of programming languages.
- II: Brief review of programming paradigms.
- III: Procedural languages.
- IV: Object-oriented languages.
- V: Functional languages.
- VI: Declarative and non-algorithmic languages.
- VII: Scripting languages.
- VIII: The effects of scalability in programming methodologies.

2. Introduction to the Translation of Languages

- I: Comparison between interpreters and compilers.
- II: Phases of language translation (lexical analysis, syntactic analysis, code generation, optimization)
- III: Dependent and independent aspects of the machine.

3. Language Translation Systems

- I: Application of regular expressions in lexical analyzers.
- II: Syntactic analysis (concrete and abstract syntax, trees of abstract syntax).
- III: Application of context-free grammars in a table-driven or recursive descending parsing.
- IV: Symbol tables administration.
- V: Generation of code by tracking a tree.
- VI: Specific operations of the architecture: selection of instructions and allocation of records.
- VII: Optimization techniques.
- VIII: The use of tools as support in the translation process and its advantages.
- IX: Program libraries and separate compilation.
- X: Construction of tools guided by the syntax.

4. Parallelism at the instructional level

- I: Processor architecture.
- II: Code programming restrictions.
- III: Basic block programming.
- IV: Global code programming.
- V: Software channeling.

5. Optimization for parallelism and locality

- I: Basic Concepts.
- II: Matrix Multiplication.
- III: Iteration Spaces.
- IV: Indexes of related arrays.
- V: Array data dependency analysis.
- VI: Parallelism search without synchronization.
- VII: Synchronization between parallel cycles.

IV. BIBLIOGRAPHY

- Aho, A. Compilers Principles, techniques and tools. Addison Wesley. 1990.
- Aho, A., Lam, M., Sethi, R., and Ullman, J. D. Compilers. Principles, techniques and tools. Addison Wesley, 2nd edition. ISBN: 10-970-26-1133-4. 2008
- A.Lemone, K. Foundations of Compilers. CECSA-Mexico. 1996.
- Appel, A. W. Modern compiler implementation in Java. Cambridge University Press, 2nd edition. 2002
- Louden, K. C. Building Compilers Principles and Practice. Posted by Thomson. 2004
- Louden, K. C. Programming Languages. Posted by Thomson. 2004
- Pratt, T. W. and V.Zelkowitz, M. Programming Languages Design and Implementation. Prentice-Hall Hispanoamericana S.A. 1998

- Teufel, B. and Schmidt, S. Foundations of Compilers. Addison Wesley Iberoamericana. 1998