UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-421	Compiler Design	6
Prerequisites	**Department**	**Semester**
COMP-211, COMP-321	Computer Science	Fall
Type of Course	**Field**	**Language of Instruction**
Elective	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Dr Ioanna Dionysiou	4th
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	None

Course Objectives:

The main objectives of the course are to:

- introduce the compilation phases
- demonstrate the application of regular expressions in lexical scanners
- examine parsing (concrete and abstract syntax, abstract syntax trees) and application of context-free grammars in recursive-descent parsing and bottom-up parsing
- discuss declarations and types
- examine run-time environments
- discuss intermediate code representations and code generation principles.

Learning Outcomes:

After completion of the course students are expected to be able to:

1. describe the various stages of the basic language translation process (lexical, parsing, code generation, optimization) and machine-dependent vs. machine-independent aspect of translation
2. recognize the underlying formal models such as finite state automata and their connection to language definition through regular expressions and grammars
3. use parsing techniques, including LL(1) and LR parsers
4. translate statements into three-address code
5. identify the properties of a variable and discuss type incompatibility
6. describe and use static vs. dynamic storage allocation and the usage of activation records to manage program modules and their data.

7. given an intermediate representation, along with symbol table information, produce a semantically equivalent target program
8. design and implement a simple language translator using automated tools, such lexical and parser generators lexx/yacc.

Course Content:

1. Overview of Compilation
2. Lexical Analysis, including regular expressions, finite automata (NFA, DFA), implementation of lexer using automated tools
3. Syntax Analysis, including context-free grammars, top-down parsing, bottom-up parsing, implementation of a parser using automated tools
4. Syntax-directed translation
5. Type Systems
6. Intermediate representations (graphical and linear)
7. Code optimization and generation

Learning Activities and Teaching Methods:

Lectures, Practical Exercises, and In-class Problem Solving Sessions

Assessment Methods:

Final Exam, Midterm Exam, and Semester Project

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Compilers: Principles, Techniques, and Tools (2nd edition)	A. V. Aho, M. Lam, R. Sethi, and J. D. Ullman	Pearson Education	2007	978-0321486813

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Engineering a Compiler (2nd Edition)	K. Cooper and L. Torczon	Morgan Kaufmann	2011	978-0120884780
Modern Compiler Design (2nd Edition)	D. Grune, K. van Reeuwijk, H. E. Bal, C. J.H. Jacobs, K. Langendoen	Springer	2012	978-1461446989