University of Nicosia, Cyprus

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

Objectives of the Course:

The main objectives of the course are to:

- present and explain the compilation phases
- discuss the application of regular expressions in lexical scanners
- discuss 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
- provide student with knowledge on run-time environments, intermediate code representations and code generation principles
- experiment with the design and implementation of a compiler for a small language

Learning Outcomes:

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

1. demonstrate 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. differentiate static vs. dynamic storage allocation and the usage of activation records to manage program modules and their data
7. produce a semantically equivalent target program, given an intermediate representation, along with symbol table information,
8. design and implement a simple language translator using automated tools, such lexical and parser generators lexx/yacc

Course Contents:

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, in-class problem solving sessions

Assessment Methods:

Homework, project, midterm exam, final exam

Required Textbooks/Reading:

Authors	**Title**	**Publisher**	**Year**	**ISBN**
Alfred V. Aho, Monica Lam, Ravi Sethi, and Je□rey D. Ullman	*Compilers: Principles, Techniques, and Tools (2nd edition)*	Pearson Education, Inc	2007	0321486811

Recommended Textbooks/Reading:

Authors	**Title**	**Publisher**	**Year**	**ISBN**
Keith Cooper and Linda Torczon	*Engineering a Compiler*	Morgan Kaufmann	2003	155860698X
John Levine, Tony Mason, and Doug Brown	*Lex and Yacc (2nd edition)*	O'Reilly Press	1992	1565920007