UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-321	Theory of Computation	6
Prerequisites	**Department**	**Semester**
COMP-270	Computer Science	Fall
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Prof. Ioanna Dionysiou	3rd
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	None

Course Objectives:

The main objectives of the course are to:

- introduce the basic theoretical principles in Computer Science
- compare and contrast the various types of finite automata
- thoroughly discuss formal definitions of programming languages and their connection with finite automata
- cover in detail Turing machines and computability
- introduce the theoretical understanding of the halting problem.
- cover time complexity theory (class P, class NP, NP-completeness)

Learning Outcomes:

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

1. apply techniques to construct finite state machines and regular expressions
2. apply techniques to design context-free languages
3. design a (non)deterministic finite-state machine to accept a specified language
4. explain how some problems have no algorithmic solution
5. analyze examples that illustrate the concept of uncomputability
6. prove that a language is in a specified class and that it is not in the next lower class.
7. apply techniques to convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs
8. analyze the Church-Turing thesis and its significance

9. discuss the Halting Problem
10. demonstrate the usage of reductions to decide if a problem is solvable or insolvable
11. analyze class P, class NP, NP-complete problems.

Course Content:

1. Automata and Languages
    a. Regular Languages
        i. Finite Automata (FA)
        ii. Deterministic FA and Nondeterministic FA
        iii. Regular Expressions and Languages
    b. Context-free Grammars and Languages
        i. Context-free Grammars
        ii. Pushdown Automata (PDAs)
        iii. Non-Context-Free Languages
2. Computability Theory
    a. The Church-Turing Thesis
        i. Turing Machines
        ii. Variants of Turing Machines
    b. Decidability
        i. Decidable Languages
        ii. Diagonalization
        iii. The Halting Problem
    c. Reducibility
        i. Reductions
3. Complexity Theory
    a. Time complexity (class P, class NP, NP-completeness)

Learning Activities and Teaching Methods:

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

Assessment Methods:

Final Exam, Midterm Exam, and Assignments

UNIC

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Introduction to Automata Theory, Languages, and Computation (3rd Ed.)	John Hopcroft, Rajeev Motwani, Jeffrey Ullman	Pearson	2006	978-0321455369

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Introduction to the Theory of Computation	William A. Goddard	Jones & Bartlett Publishers	2008	978-0763741259
Introduction to the Theory of Computation (3rd Ed.)	Michael Sipser	Course Technology	2012	978-1133187813