University of Nicosia, Cyprus

Course Code	**Course Title**	**ECTS Credits**
COMP-321	Theory of Computation	6
Department	**Semester**	**Prerequisites**
Computer Science	Fall, Spring	COMP-211
Type of Course	**Field**	**Language of Instruction**
Required	Computer Science	English
Level of Course	**Year of Study**	**Lecturer(s)**
1st Cycle	3rd	Dr Florent Domenach
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:

- be familiar with the basic theoretical principles in Computer Science
- know various types of finite automata
- be familiar with formal definitions of programming languages and their connection with finite automata
- have learnt material on Turing machines and computability
- have a deeper theoretical understanding of algorithmic complexity classes.

Learning Outcomes:

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

1. Discuss the concept of finite state machines.
2. Explain context-free grammars.
3. Design a deterministic finite-state machine to accept a specified language.
4. Explain how some problems have no algorithmic solution.
5. Provide examples that illustrate the concept of uncomputability.
6. Determine a language's location in the Chomsky hierarchy (regular sets, context-free, context-sensitive, and recursively enumerable languages).
7. Prove that a language is in a specified class and that it is not in the next lower class.
8. Convert among equivalently powerful notations for a language, including among DFAs, NFAs, and regular expressions, and between PDAs and CFGs.
9. Explain at least one algorithm for both top-down and bottom-up parsing.
10. Explain the Church-Turing thesis and its significance.
11. Define the classes P and NP.
12. Explain the significance of NP-completeness.
13. Prove that a problem is NP-complete by reducing a classic known NP-complete problem to it.

Course Contents:

1. Automata and Languages

- Finite-state machines
- Context-free grammars

- Tractable and intractable problems
- Uncomputable functions
- The halting problem
- Implications of uncomputability
- Deterministic finite automata (DFAs)
- Nondeterministic finite automata (NFAs)
- Equivalence of DFAs and NFAs
- Regular expressions
- The pumping lemma for regular expressions
- Push-down automata (PDAs)
- Relationship of PDAs and context-free grammars
- Properties of context-free grammars

2. Computability Theory

- Turing machines
- Nondeterministic Turing machines
- Sets and languages
- Chomsky hierarchy

3. Complexity Theory

- The Church-Turing thesis
- Definition of the classes P and NP
- NP-completeness (Cook's theorem)
- Standard NP-complete problems
- Reduction techniques

Learning Activities and Teaching Methods:

Lectures, Practical Exercises and Assignments.

Assessment Methods:

Homework, Mid-Term, Final Exam.

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Wayne Goddard	Introducing the theory of computation	Jones & Bartlett Publishers	2008	9780763741259

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Harry Lewis Christos Papadimitriou	Elements of the theory of Computation	Prentice Hall, 2nd edition	1997	0132624788
Michael Sipser	Introduction to the theory of Computation	Brooks Cole, 1st edition	1996	053494728
John Hopcroft, Rajeev Motwani Jeffrey Ullman	Introduction to Automata Theory, Languages, and Computation	Addison-Wesley	2008	0-321-45536-3