UTEC

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

University of Engineering and Technology
School of Computer Science
Syllabus of Course
Academic Period 2018-II

1. **Code and Name:** CS1D01. Discrete Structures I
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HP;
4. **Professor(s)**

Lecturer

- José Fiestas
  - PhD in Natural Science, Heidelberg, Germany, 2006.

Meetings after coordination with the professor

5. **Bibliography**

[Epp10] Susanna S. Epp. *Discrete Mathematics with Applications*. 4 ed. Brooks Cole, 2010.

[Gri03] R. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction*. 5 ed. Pearson, 2003.

[Ros07] Kenneth H. Rosen. *Discrete Mathematics and Its Applications*. 7 ed. Mc Graw Hill, 2007.

[Sch12] Edward R. Scheinerman. *Mathematics: A Discrete Introduction*. 3 ed. Brooks Cole, 2012.

6. **Information about the course**

(a) **Brief description about the course** Discrete structures provide the theoretical foundations necessary for computation. These fundamentals are not only useful to develop computation from a theoretical point of view as it happens in the course of computational theory, but also is useful for the practice of computing; In particular in applications such as verification, cryptography, formal methods, etc.

(b) **Prerrequisites:** None

(c) **Type of Course:** Mandatory

(d) **Modality:** Face to face

7. **Specific goals of the Course**

- Apply Properly concepts of finite mathematics (sets, relations, functions) to represent data of real problems.
- Model real situations described in natural language, using propositional logic and predicate logic.
- Determine the abstract properties of binary relations.
- Choose the most appropriate demonstration method to determine the veracity of a proposal and construct correct mathematical arguments.
- Interpret mathematical solutions to a problem and determine their reliability, advantages and disadvantages.
- Express the operation of a simple electronic circuit using Boolean algebra.

8. **Contribution to Outcomes**

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

a) An ability to apply knowledge of mathematics, science. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

9. Competences (IEEE)

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

C1. An intellectual understanding and the ability to apply mathematical foundations and computer science theory.⇒ **Outcome a**

C20. Ability to connect theory and skills learned in academia to real-world occurrences explaining their relevance and utility.⇒ **Outcome j**

10. List of topics

1. Sets, Relations, and Functions
2. Basic Logic
3. Proof Techniques
4. Digital Logic and Data Representation

11. Methodology and Evaluation

Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Lab Sessions:
In order to verify their competences, several activities including active learning and roleplay will be developed during lab sessions.

Oral Presentations:
Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

Reading:
Throughout the course different readings are provided, which are evaluated. The average of the notes in the readings is considered as the mark of a qualified practice. The use of the UTEC Online virtual campus allows each student to access the course information, and interact outside the classroom with the teacher and with the other students.

Evaluation System:
The final note F Depends on several intermediate notes.

- The note T is the average, rounded up, of short exams over nine points. This note is individual.
- The note P is the average, rounded up, of the workbooks on nine points. This note is group.
- The note E is the note of the problems of effort,which is an integer between zero and two. This note is individual.

To calculate the final grade F You should see student performance in three bands of performance, high performance, medium performance and low performance.

High performance : Si $\min(T, P) \geq 7$ then $F = T + P + E$.

Medium performance: Si $\min(T, P) < 7$ y $\min(T, P) \geq 4$ then $F = T + P$.

Low performance: If $\min(T, P) < 4$ then $F = 2 * \min(T, P)$.

To pass the course you must obtain 11 or more in the final grade F.

12. Content

Unit 1: Sets, Relations, and Functions (13)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Explain with examples the basic terminology of functions, relations, and sets [Assessment] • Perform the operations associated with sets, functions, and relations [Assessment] • Relate practical examples to the appropriate set, function, or relation model, and interpret the associated operations and terminology in context [Assessment]	• Sets – Venn diagrams – Union, intersection, complement – Cartesian product – Power sets – Cardinality of finite sets • Relations – Reflexivity, symmetry, transitivity – Equivalence relations, partial orders • Functions – Surjections, injections, bijections – Inverses – Composition
Readings : [Gri03], [Ros07]	

Unit 2: Basic Logic (14)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Convert logical statements from informal language to propositional and predicate logic expressions [Usage] • Apply formal methods of symbolic propositional and predicate logic, such as calculating validity of formulae and computing normal forms [Usage] • Use the rules of inference to construct proofs in propositional and predicate logic [Usage] • Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (eg, program correctness), database queries, and algorithms [Familiarity] • Apply formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles [Usage] • Describe the strengths and limitations of propositional and predicate logic [Usage]	• Propositional logic • Logical connectives • Truth tables • Normal forms (conjunctive and disjunctive) • Validity of well-formed formula • Propositional inference rules (concepts of modus ponens and modus tollens) • Predicate logic – Universal and existential quantification • Limitations of propositional and predicate logic (e.g., expressiveness issues)
Readings : [Ros07], [Gri03]	

Unit 3: Proof Techniques (14)	
Competences Expected: C1,C20	
Learning Outcomes	**Topics**
• Identify the proof technique used in a given proof [Assessment] • Outline the basic structure of each proof technique (direct proof, proof by contradiction, and induction) described in this unit [Usage] • Apply each of the proof techniques (direct proof, proof by contradiction, and induction) correctly in the construction of a sound argument [Usage] • Determine which type of proof is best for a given problem [Assessment] • Explain the parallels between ideas of mathematical and/or structural induction to recursion and recursively defined structures [Familiarity] • Explain the relationship between weak and strong induction and give examples of the appropriate use of each [Assessment] • State the well-ordering principle and its relationship to mathematical induction [Familiarity]	• Notions of implication, equivalence, converse, inverse, contrapositive, negation, and contradiction • The structure of mathematical proofs • Direct proofs • Disproving by counterexample • Proof by contradiction • Induction over natural numbers • Structural induction • Weak and strong induction (i.e., First and Second Principle of Induction) • Recursive mathematical definitions • Well orderings
Readings : [Ros07], [Epp10], [Sch12]	

<table>
<tr><td colspan="2">Unit 4: Digital Logic and Data Representation (19)</td></tr>
<tr><td colspan="2">Competences Expected: C1,C20</td></tr>
<tr><td>Learning Outcomes</td><td>Topics</td></tr>
<tr><td>

Explain the importance of Boolean algebra as a unification of set theory and propositional logic [Assessment].
Know the algebraic structures of reticulum and its types [Assessment].
Explain the relationship between the reticulum and the ordinate set and the wise use to show that a set is a reticulum [Assessment].
Know the properties that satisfies a Boolean algebra [Assessment].
Demonstrate if a terna formed by a set and two internal operations is or not Boolean algebra [Assessment].
Find the canonical forms of a Boolean function [Assessment].
Represent a Boolean function as a Boolean circuit using logic gates [Assessment].
Minimize a Boolean function. [Assessment].

</td><td>

Partial orders and Partially ordered sets.
Extreme elements of a partially ordered set.
Reticles: Types and properties.
Boolean algebras.
Boolean Functions and Expressions.
Representation of Boolean Functions: Normal Disjunctive and Conjunctive Form.
Logical Doors.
Circuit Minimization.

</td></tr>
<tr><td colspan="2">Readings : [Ros07], [Gri03]</td></tr>
</table>