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

1. **Code and Name:** CS1D02. Discrete Structures II
2. **Credits:** 4
3. **Hours of theory and Lab:** 2 HT; 4 HP;
4. **Professor(s)**

**Lecturer**

- Yamilet Rosario Serrano Llerena
  - PhD in Computer Science, National University of Singapore, Singapore, 2018.
- José Miguel Renom
  - PhD in Math, Universidad Simón Bolívar, Venezuela, 2016.

Meetings after coordination with the professor

5. **Bibliography**

[Gri97] R. Grimaldi. *Matemáticas Discretas y Combinatoria*. Addison Wesley Iberoamericana, 1997.

[Joh99] Richard Johnsonbaugh. *Matemáticas Discretas*. Prentice Hall, México, 1999.

6. **Information about the course**

(a) **Brief description about the course** In order to understand the advanced computational techniques, the students must have a strong knowledge of the Various discrete structures, structures that will be implemented and used in the laboratory in the programming language..

(b) **Prerrequisites:** CS1D01. Discrete Structures I. ($1^{st}$ Sem)

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

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

7. **Specific goals of the Course**

- That the student is able to model computer science problems using graphs and trees related to data structures
- That the student apply efficient travel strategies to be able to search data in an optimal way

8. **Contribution to Outcomes**

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

**b)** An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

**i)** An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

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

**b)** An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

**i)** An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Familiarity**)

9. **Competences (IEEE)**

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

**C5.** Ability to implement algorithms and data structures in software.$\Rightarrow$ **Outcome b**

**CS2.** Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.$\Rightarrow$ **Outcome i**

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

**C5.** Ability to implement algorithms and data structures in software.$\Rightarrow$ **Outcome b**

**CS2.** Identify and analyze criteria and specifications appropriate to specific problems, and plan strategies for their solution.$\Rightarrow$ **Outcome i**

## 10. List of topics

1. Basics of Counting
2. Graphs and Trees

## 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: Basics of Counting (45)**

**Competences Expected: C1**

**Learning Outcomes**

- Apply counting arguments, including sum and product rules, inclusion-exclusion principle and arithmetic/geometric progressions [Familiarity]
- Apply the pigeonhole principle in the context of a formal proof [Familiarity]
- Compute permutations and combinations of a set, and interpret the meaning in the context of the particular application [Familiarity]
- Map real-world applications to appropriate counting formalisms, such as determining the number of ways to arrange people around a table, subject to constraints on the seating arrangement, or the number of ways to determine certain hands in cards (eg, a full house) [Familiarity]
- Solve a variety of basic recurrence relations [Familiarity]
- Analyze a problem to determine underlying recurrence relations [Familiarity]
- Perform computations involving modular arithmetic [Familiarity]

**Topics**

- Counting arguments
  - Set cardinality and counting
  - Sum and product rule
  - Inclusion-exclusion principle
  - Arithmetic and geometric progressions
- The pigeonhole principle
- Permutations and combinations
  - Basic definitions
  - Pascal's identity
  - The binomial theorem
- Solving recurrence relations
  - An example of a simple recurrence relation, such as Fibonacci numbers
  - Other examples, showing a variety of solutions
- Basic modular arithmetic

**Readings :** [Gri97]

**Unit 2: Graphs and Trees (45)**

**Competences Expected: C1**

**Learning Outcomes**

- Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each type of graph/tree [Familiarity]
- Demonstrate different traversal methods for trees and graphs, including pre, post, and in-order traversal of trees [Familiarity]
- Model a variety of real-world problems in computer science using appropriate forms of graphs and trees, such as representing a network topology or the organization of a hierarchical file system [Familiarity]
- Show how concepts from graphs and trees appear in data structures, algorithms, proof techniques (structural induction), and counting [Familiarity]
- Explain how to construct a spanning tree of a graph [Familiarity]
- Determine if two graphs are isomorphic [Familiarity]

**Topics**

- Trees
  - Properties
  - Traversal strategies
- Undirected graphs
- Directed graphs
- Weighted graphs
- Spanning trees/forests
- Graph isomorphism

**Readings :** [Joh99]