Course Definition: Discrete Structures

1- Basic Information:

Course Name	Discrete Structures
Course ID	DSR
Contact Hours (Registered Sessions)	2 hours
Contact Hours (Synchronized Sessions)	18
Mid Term Exam	2 hours
Exam	1.15
Registered Sessions Work Load	6
Synchronized Session Work Load	36
Credit Hours	4

2- Pre-Requisites:

Course	ID
Programming 1	PRG 1

3- Course General Objectives:

The course aims to implantation the basic concepts in Discrete Structures in developing the student's Information Technology and mathematical abilities.

The student is able to design logical and electronic circuits depending on the logic and the Boolean Algebra and to ensure the accuracy of the output mathematically and in conformity with the design, as well as acquiring a scientific basis to develop and understand the work of compliers and mechanisms of the programming languages and text processors.

Students can design algorithms to solve various problems and make measurements on them. In general, the course builds a solid foundation for students to use theoretical knowledge in practice.

4- Intended Learning Outcomes (ILO):

Code	Intended Learning Outcomes
ILO1	Enhancing the use of mathematical logic and methods of proof for students in different fields of information Technology
ILO2	Designing of electronic and logic circuits and mathematical certainty of mechanism of work and results
ILO3	Development of some algorithms based on the concepts of Discrete Structures
ILO4	Understanding the mechanism of Compliers, text editors, language design and Building **Automata**.
ILO5	Development in the construction of advanced data structures
ILO6	The movement Form Propositional logic to **Predicate logic** ,the qualification to study the foggy logic and artificial intelligence
ILO7	Using **Graph** theory and Trees in designs and solutions

5- Course Syllabus (18 hours of total synchronized sessions)

- Logic & Proof
- Boolean Algebra
- Recursive & induction
- predicate logic
- Algorithms
- Graph
- Trees
- **Automata & FL**

- **RS:** Recorded Sessions; **SS:** Synchronized Sessions;

ILO	Course Syllabus	RS	SS	Type	Additional Notes
ILO1	• **Logic & Proof**		١.٥	assignments	Exercises

ILO2	• Boolean Algebra		٣	assignments	Exercises
ILO3	• Algorithms - Recursive & induction	١	٣	assignments	Exercises
ILO4	• Automata & FL		٣	assignments	Exercises
ILO5	• Graph – Trees		٣	assignments	Exercises
ILO6	• Predicate Logic	١	١.٥	assignments	Exercises
ILO7	• Graph – Trees		٣	assignments	Exercises

6- Assessment Criteria (Related to ILOs)

ISC	Interactive Synchronized Collaboration		Ex	Exams		Rpt	Reports
PF2F	Presentations and Face-to-Face Assessments		PW	Practice Work			

ILO Code	ILO	Intended Results	Assessment Type				
			ISC	PW	Ex	PF2F	Rpt
ILO1	Gain skill and knowledge in each of the following: • Mathematical Logic • Sentential (Propositional) Calculus • Connectivity Tools • Logical equivalence • The principle of	Solve exercises and questions. And use these skills in programming and algorithms	✓		✓		

	compensation for credits • Methods of proof						
ILO2	• The structure of Boolean Algebra • Calculation in Boolean algebra • Rules relating to Boolean standards • Arrangement relations in the Boolean algebra • Partial Boolean algebra • Free groups of Boolean algebra • Groups generated in Boolean algebra • The basis for the Boolean algebra • Study of Boolean with several variables • designing Logical circuits	Solving exercises and designing logical circuits and Using mathematical theories in the field of Boolean algebra in matters related to Hardware	✓		✓		✓
ILO3	• Format of the Theorem or Proposition • Induction Proofs • Induction Proof on Recursive Procedure • Recursion • Recursive Trees • Time complexity • Algorithms design techniques	Make the student able to compare algorithms, complexity and design methods, and Preview the recursive trees that will be used in different data structures and algorithms	✓		✓		✓

ILO4	• Automata of inevitable end • Automata of Infinite Ending • Automata that ending with € move • Systematic expression • Regular languages • Applications	The design and implementation Automata of certain conditions, the acceptance of language and system expressions, the use of automation in compliers and text processors, and the design of some hardware based on the concept of automation	✓		✓		✓
ILO5	• Introduction and definitions in the Graph • Useage of the Graph • Types of Graph • Movement on the Graph • Types of trees • Binary trees • Search trees	The ability to use the theory of the Graph and trees in solving general issues and use them in algorithms and the development of the concepts of representation of data structures of this type in the memories and enable the algorithms of Movement and modification of the structures and	✓		✓		✓

		reading from them					
ILO6	• The movement Form Propositional logic to **Predicate logic** • Concepts and rules in the **Predicate logic** • Convert language sentences to documents • Allocation and generalization and inclusion • Robot control example	Make the student able to expanded the mathematical logic to Predicate logic and use the documents to solve the exercises, especially with artificial intelligence and to qualify the student to understand the foggy logic	✓		✓		✓
ILO7	• Trees representation • Graph representation • Operations on the Graph • The relationship between the Graph and the trees	Solving exercises using the Graph and trees in the development of advanced algorithms and moving the problem from the case of text to the mathematical design of the intermittent structure and finding solutions to the problem	✓		✓		✓

7- Practice Tools:

Tool Name	Description
N/A	

8- Main References

1- Susanna S. Epp. ,Discrete Mathematics with Applications. DePaul University, 2015 edition.

2- Daniel I.A. Cohen, "Introduction to Computer Theory", John Wiley & Sons, Inc, 2013 edition

9- Additional References

1- Thomas H. Cormen, " Introduction to Algorithms", The MIT Press, 2000

2- Sara Baase and Allen Van Gelder, " Computer Algorithms : Introduction to Design and Analysis", Pearson Education, Inc, 2002

3- Kenneth A. Ross and Charles R.B. Wright, Discrete Mathematics, Fifth Edition , University of Oregon 2003