

Discrete Logic for Computers

2023-2024 Catalog

[ARCHIVED CATALOG]

CSCI 105 - Discrete Logic for Computers

PREREQUISITES: [CSCI 101 - Computer Science I](#) or ([SDEV 140](#) and [CSCI 179](#))

PROGRAM: Computer Science

CREDIT HOURS MIN: 3

LECTURE HOURS MIN: 3

DATE OF LAST REVISION: Fall 2020

This course introduces students to discrete mathematical concepts including reasoning and proof, especially with the discrete phenomenon often used in the field of Computer Science. Students will learn the applicable mathematical vocabulary and its correct usage. Other topics include sets, functions, notation, proofs, proof techniques, relations, induction, counting and countability, probability, and partitions.

MAJOR COURSE LEARNING OBJECTIVES: Upon successful completion of this course the student will be expected to:

1. Interpret and write proofs, both directly and through contradiction.
2. 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
3. Apply proof techniques (direct proof, proof by contradiction, and induction) in the construction of a sound argument.
4. Demonstrate usage of functions, relations and sets
5. Explain the Pigeonhole Principle, cardinality and "countability."
6. Describe Boolean values and perform standard operations on them.
7. Explain and apply de Morgan's laws.
8. Identify and implement logical connectives, truth tables, normal forms and validity.
9. Discuss the concepts of digital logic: logic gates, flip-flops, counters, circuit minimization.
10. Demonstrate different aspects of descriptive statistics, including methods of collecting data, frequency distribution, measures of central tendency, variation and position, and use of z-scores.
11. Describe the basics of counting including counting arguments, permutations and combinations and binomial coefficients.
12. Describe how symbolic logic can be used to model real-life situations or applications, including those arising in computing contexts such as software analysis (e.g., program correctness), database queries, and algorithms.
13. Convert logical statements from informal language to propositional and predicate logic expressions.
14. Describe the strengths and limitations of propositional and predicate logic.
15. Apply the rules of inference to construct proofs in propositional and predicate logic.
16. Explain the relationship between weak and strong induction and give examples of the appropriate use of each.
17. Explain the basic terminology of graph theory and discuss real-world problems solvable by using graphs.



COURSE CONTENT: Topical areas of study include -

- Boolean algebra
- Descriptive statistics
- Digital logic
- Elementary number theory

- Functions
- Fundamentals of counting
- Graph Theory
- Predicate logic
- Proof techniques
- Propositional logic
- Recurrence relations
- Relations
- Sets

[Course Addendum - Syllabus \(Click to expand\)](#)

