1. *Course number and name*
   CSC 230: Discrete Mathematical Structures for Computer Science

2. *Credits and contact hours*
   3 credits
   Contact hours: 150 minutes of lecture sessions /week

3. *Instructor's or course coordinator's name*
   Course coordinator: Kazunori Okada, Associate Professor of Computer Science

4. *Text book, title, author, and year*

   Discrete Mathematics for Computer Science, Haggard G, Schlipf J, Whitesides S. *Discrete Mathematics and Its Application, 6th Edition*, Kenneth Rosen

   a. *other supplemental materials*
      Lecture Slides

5. *Specific course information*
   - *brief description of the content of the course (catalog description)*

     Review of set algebra, relations and functions; permutations; propositional logic; proof techniques; introduction to graph theory; infinite sets; applications to computer science.

   - *prerequisites or co-requisites*

     CSC 210 with grade of C or better; MATH 227 with grade of C or better or must be taken concurrently.

   - *indicate whether a required, elective, or selected elective course in the program*
     Required for Computer Engineering.

6. *Specific goals for the course*
   a. *specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.*

      At the end of this course students will be able to

- Solve mathematical problems with discrete objects by identifying and using appropriate discrete mathematical methods
- Understand and create proofs of simple concrete problems
- Use correct logical inference to conduct deductive proofs
- Define and analyze functions
- Choose an appropriate method for counting discrete objects
- Solve the shortest-path problem by using Dijkstra's algorithm

- Analyze time complexity of an algorithm given in a pseudo code by deriving its big Oh

b. *explicitly indicate which of the student outcomes listed in Criterion 3 or any other outcomes are addressed by the course.*
   Course addresses ABET Student Outcome(s): a, b, c, e, j, k.

7. *Brief list of topics to be covered*
   - **Sets, relations, and functions:** Sets (set operations, complements, Venn diagrams, membership table, Cartesian products, power sets, cardinality); Relations (reflexivity, symmetry, transitivity, compositions, closures, equivalence relations, equivalence class, partitions); Functions (injections, surjections, bijections, inverses, compositions);
   - **Basic logic:** Propositional logic; Logical connectives; Truth tables; Normal forms (conjunctive and disjunctive); Validity; Predicate logic; Logical inference for deductive proofs; Modus ponens and modus tollens; Universal and existential quantification;
   - **Proof techniques:** The structure of formal proofs; Direct proofs; Proof by Cases; Proof by counterexample; Proof by contraposition; Proof by contradiction; Mathematical induction; Strong induction; Recursive mathematical definitions; Deductive proof;
   - **Basics of counting:** Counting arguments; Sum and product rule; Inclusion-exclusion principle; Arithmetic and geometric progressions; The pigeonhole principle; Set countability; Permutations and combinations; Pascal's identity; The binomial theorem; Fibonacci numbers;
   - **Trees and Graphs:** Graphs; Graph operations; Graph types; Handshaking theorem; Graph representation (adjacency matrix and adjacency list); Isomorphism; Shortest path problem; Dijkstra's algorithm; Trees; Spanning trees; Traversal strategies;
   - **Introduction to Analysis of Algorithms:** Big Oh Notations; Solving recurrence relations; Computability theory (Halting problem); Number theory; Prime number; Modular arithmetic; Euclidean algorithm