New Course Proposal

Submit completed form electronically

1. **Course prefix and course number:** CS - 250

2. **Course title:** Discrete Structures for Computer Science

3. **Abbreviated title for class schedule** (30 characters or less): Discrete Structures

4. **Credit hours:** 4
   (note: if credits are variable, list range of credits (e.g. 1-8 credits)

5. **Catalog description:** An introduction to discrete mathematics for computer science applications, with an emphasis on rigorous proof methods. Topics include logic, functions, sequences, sets, induction, recursion, combinatorics, and graph theory.

6. **Prerequisites (*to add each additional prerequisite, start a new line*):**
   (*See attached Note for samples***)**

   A. **(course prefix, (space) and number)** Mth 112 with C- or better (or appropriate SOU placement level)
   B. (course prefix, (space) and number) or or or or

   C. (course prefix, (space) and number) or or or or

7. **Co-requisites (including labs, if any):**
   A. (course prefix, (space) and number) or or or or

8. **Major/Class restrictions: Please indicate any class or major restrictions:**

9. **Is course repeatable? Yes No X If Yes, list maximum credits:**

10. **Labs requirements: If course includes a lab: # of hours lecture: ; # of hours lab:**

11. **Fees: List any course fees:**

12. **Grade Mode: Graded only: Pass/No Pass only: Option: X**

13. **CIP Code: Six-digit CIP code** (check with your Division Director): 11.0101

14. **Special qualifications; Is course proposed for (yes/no):**
    A. University Studies? If yes, list Strand(s)

    B. Honors?

15. **Cross-listing: List any cross-listing (and please complete the Cross-list proposal form at https://inside.sou.edu/provost/curriculum.html):** and and and and

16. Strategic justification for proposed course:

A. **Rationale**: What is the overall strategic rationale for offering this course? Preparing students to CS-418 (Theory of Computation) and improve and simplify our current prerequisite matrix.

To improve the mathematical and problem solving background needed for upper division course work, particularly CS 418. This will allow CS 418 to much better explore the breadth of content and also provides to the student the logic reasoning and improve student background in algorithms; Furthermore, we are examining our prerequisite structure to remove unnecessary roadblocks for students. Introducing this course is the first step in this process.

B. **Alignment**:

1. How does this course align with the unit's mission plan? Our new course aims to providing scientific background and foundations of Computer Science to our student. This sense we intend to create an environment to support creativity, critical thinking, discovering, and cultivate bold ideas and actions.

2. How does the course fit into the rest of the unit's curriculum? Offering this course, we will prepare students to the Theory of Computation and Compilers. However, we will be able to shorten our programming sequence and reduce pressure for trailing sessions by cutting prerequisites.

C. **Enrollment**: What is the new course's estimated enrollment each time it is offered over a three-year period? Year 1 14 - 16 ; Year 2 22 - 26 ; Year 3 28 - 32

D. **Resource evaluation:** What resources – faculty, equipment, lab space, etc. -- will be needed to offer this course and how will those resources be obtained?

1. *Faculty*:
   a. Who will teach the course? CS and MATH faculty
   b. Evaluate unit's faculty availability and/or needs and the impact on other teaching obligations. May reduce the amount of GE/Service taught by full-time faculty
   c. If additional faculty members are needed, how will that need be met? TxT or support from Math

2. *Facilities*: Cite any additional need for classrooms, equipment or lab space; explain how that need(s) will be met. None

3. *Other*:

a. Are Hannon Library resources sufficient to meet the needs of this course?
yes Yes

b. Are any other resources needed to support this course? No
If so, please explain how they will be obtained. -

E. **External impact**:

1. What is the expected effect of this course on existing programs elsewhere in the university? Minimal if any. Minimal, if any

NOTE: Please document your contact with other academic programs which may be affected by this new course and the response you received.

2. Will any of your prerequisites affect other academic programs? There is no impact in other programs. CS students are already taking Mth 111/112 to meet prerequisites for the Mth 251/252 requirement.

NOTE: Please document your contact with other academic programs which may be affected by this new course and the response you received.

17. Syllabus (condensed)

*(Attach an accompanying, condensed syllabus, which should include the following items. Schedules and similar details are **not** required.)*

A. Course description (same as Catalog description, above)

An introduction to discrete mathematics for computer science applications, with an emphasis on rigorous proof methods. Topics include logic, functions, sequences, sets, induction, recursion, combinatorics, and graph theory.

B. Learning objectives of the course

The main goal of this course is that students obtain those skills in discrete mathematics and logic that are used in the study and practice of computer science. Upon the successful completion of this course students will be able to:

Describe basic properties of sets, bags, tuples, relations, graphs, trees, and functions.
Perform traversals of graphs and trees; construct simple functions by composition of known functions; determine whether simple functions are injective, surjective, or bijective; and classify simple functions by rate of growth.
Describe the concepts of countable and uncountable sets, and apply the diagonalization method to construct elements that are not in certain countable sets.
Construct inductive definitions for sets, construct grammars for languages (sets of strings), and construct recursive definitions for functions and procedures.
Determine whether a binary relation is reflexive, symmetric, or transitive and construct closures with respect to these properties.
Construct a topological sort of a partially ordered set and determine whether a partially ordered set is well-founded.
Use elementary counting techniques to count simple finite structures that are either ordered or unordered, to count the worst case number of comparisons and, with discrete probability, to count the average number of comparisons for simple decision trees.
Find closed form solutions for simple recurrences using the techniques of substitution, cancellation, and generating functions.
Demonstrate standard proof techniques and the technique of inductive proof by writing short informal proofs about simple properties of numbers, sets, and ordered structures.

C. Required texts

*Susanna S. Epp. **Discrete Mathematics with Applications.** Cengage Learning; 5 edition (January 1, 2019). ISBN-10: 1337694193*

D. Course format

The course is devoted to problem solving techniques of discrete mathematics. The exercises and tests require problem analysis to find out which tools of discrete mathematics are needed to solve a problem. Therefore, the course format is based on interactive lectures, where instructors provides not only techniques, demonstrations and proofs, but also, exercises to train and challenge students, in a sense to develop their problem solving reasoning.

E. Other – any other relevant materials needed to explain the goals and teaching methods of this course.

Approvals:

Signature of Division Director	Date

4/29/16