

UNDERGRADUATE CURRICULUM CHANGE MEMO

Date:

From: Luc Longpré, Undergraduate Program Director, Computer Science

Through: Dr. Ann Gates, Chair, Dept. of Computer Science

Through: Dr. Louis Everett, Chair, Curriculum Committee, College of Engineering

Through: Dr. Theresa Maldonado, Dean, College of Engineering

To: Dr. Jose Maria Herrera, Chair Undergraduate Curriculum Committee

Proposal Title: Computer Science curriculum changes for the Fall 2020 catalog

The Department of Computer Science proposes the following changes to the Undergraduate Curriculum and to the Bachelor of Science in Computer Science degree plan:

- 1) The department proposes to add new courses: CS 1191 (Special Topics in Computing) and CS 1291 (Special Topics in Computing.)

Rationale. The department already offers CS 1190 and CS 1290, Special Topics in Computing at the freshman level. These courses can count (with restrictions) towards the technical electives in the BS in CS. The department also needs the possibility of teaching Special Topics courses at the lower division level that should not count towards the technical electives, such as pilot courses and/or courses that don't contain a substantial amount of technical material.

- 2) The department proposes to add new courses: CS 1110 (Introduction to Problem Solving,) CS 1120 (Computational Thinking in Problem Solving) and CS 2210 (Algorithmic Thinking in Problem Solving.) and include the three courses as eligible towards their technical electives:

Rationale. The Problem Solving courses were developed to address industry's need for improved problem-solving skills, incorporating consistent, deep collaboration with Google technical staff. The intent is to instill complementary problem solving, computational thinking skills, and logical reasoning needed to succeed in computer science, and make this content available across different student populations at various stages in their academic pathways. Advanced problem solving prepares students for competitive interviews. The courses create opportunities to learn across academic levels, and create new student

communities, mentorship opportunities, and social connections to support retention.

- 3) The department proposes to add new courses: CS 2101 (Discrete Structures I) and CS 2202 (Discrete Structures II.)

Rationale. The two courses have been offered as special topics every semester starting Fall 2018 as an alternate prerequisite for courses requiring MATH 2300 (Discrete Mathematics.) The courses focuses on applications of discrete structures that are central to Computer Science. The course CS 2101 contains material not dependent on MATH 1411 (Calculus I). This allows students to take the one credit course along with their first CS required course and start earlier to develop their mathematical maturity. The CS department has developed the course descriptions and outcomes in consultation with the Math department.

- 4) The department proposes to add new course: CS 4175 (Parallel Computing) as a required course for the BSCS degree.

Rationale. The area of Parallel and Distributed Computing has been added as a new competency required for accrediting computing programs by the Computing Accreditation Commission (CAC) of the Accreditation Board for Engineering and Technology (ABET). The current BSCS curriculum lacks a course focused on this area, and adding this new course will ensure that the BSCS program meets CAC/ABET accreditation criteria.

- 5) The department proposes to update the list of courses that can count as technical electives in the BS in CS degree plan.

Rationale:

The list needs to be updated with respect to new courses. For ABET accreditation purpose, the degree plan needs 12 credit hours of junior/senior level courses, so we need to limit the freshman/sophomore courses that counts towards technical electives to 3 hours. In addition, the department requires the majority of the technical elective courses taken by the students in the BS in CS to be junior/senior courses with well-defined course outcomes.

Current catalog:

Technical Electives: 15 hours from the following: CS 1190, CS 1290, CS 3000 or 4000 level course. No more than six credit hours of CS xx90, CS 4181, CS 4371, CS 4x73, CS 4392 and/or CS 4393 (in any combination) can count for technical electives.

New catalog:

Technical Electives: 15 hours from the following: CS 1110, CS 1120, CS 2210, CS 1190, CS 1290, CS 3000 or 4000 level course. No more than three credit hours of CS 1xxx and CS 2xxx can count for technical electives. No more than six credit hours of CS 1xxx, CS 2xxx, CS 4390, CS 4181, CS 4371, CS 4x73, CS 4392 and/or CS 4393 (in any combination) can count for technical electives.

- 6) The department proposes to modify some course prerequisites.

Rationale. While MATH 2300 (Discrete Math) is a prerequisite for CS 2302 at UTEP, it is not at other community colleges or universities, so we need to explicitly require MATH 2300 (or equivalently the new courses CS 2101 and CS 2202) as prerequisite for several courses at the junior/senior level. Also, remove CS 2402 as an alternate prerequisite, as the course has not been taught for over 10 years. The new courses (CS 2101 and CS 2202) should also be added as a MATH 2300 alternate prerequisite.

- 7) The department proposes to modify the CS 1301 course description.

Rationale. The course objectives have not changed, but the new description fixes some typos and is more consistent with other CS course descriptions.

CURRICULUM CHANGE PROPOSAL

APPROVAL PAGE

Proposal Title: Computer Science curriculum changes for the Fall 2020 catalog

College: Engineering

Department: Computer Science

DEPARTMENT CHAIR

I have read the enclosed proposal and approve this proposal on behalf of the department.



11/4/19

Signature

Date

COLLEGE CURRICULUM COMMITTEE CHAIR

I have read the enclosed documents and approve the proposal on behalf of the college curriculum committee.



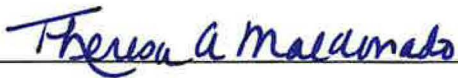
11/8/19

Signature

Date

COLLEGE DEAN

I have read the enclosed documents and approve the proposal on behalf of the college. I certify that the necessary funds will be allocated by the college in support of this proposal.



11/19/2019

Signature

Date

Graduate Council/Undergraduate Curriculum Committee

Council Action:

Approved

Returned to the College

Date of Action Report:

Signature, Chairman

Date

COURSE ADD

All fields below are required

College : Engineering

Department : Computer Science

Rationale for adding the course:

This course (with sequel CS 2202) is an alternate prerequisite of MATH 2300 Discrete Mathematics for CS upper level courses. The material aligns with MATH 2300, but is more focused to the Computer Science students' interests.

All fields below are required

Subject Prefix and # CS 2101

Title (29 characters or fewer): Discrete Structures I

Dept. Administrative Code : 720

[CIP Code](#) 110701

Departmental Approval Required Yes No

Course Level UG GR DR SP

Course will be taught: Face-to-Face Online Hybrid

How many times may the course be taken for credit? (Please indicate 1-9 times): 1

Should the course be exempt from the "Three Repeat Rule?" Yes No

Grading Mode: Standard Pass/Fail Audit

Description (600 characters maximum):

The purpose of this course is to learn to use discrete structures that are central to computer science. In particular, this course provides the foundation for and practice in analyzing computer algorithms concerning both correctness and performance, by introducing logic, proofs, sets, relations, functions, and counting.

Contact Hours (per week): 1 Lecture Hours Lab Hours Other

Types of Instruction (Schedule Type): Select all that apply

- | | | | |
|---------------------------------------|-------------------|----------------------------|-------------------------------------|
| <input checked="" type="checkbox"/> A | Lecture | <input type="checkbox"/> H | Thesis |
| <input type="checkbox"/> B | Laboratory | <input type="checkbox"/> I | Dissertation |
| <input type="checkbox"/> C | Practicum | <input type="checkbox"/> K | Lecture/Lab Combined |
| <input type="checkbox"/> D | Seminar | <input type="checkbox"/> O | Discussion or Review (Study Skills) |
| <input type="checkbox"/> E | Independent Study | <input type="checkbox"/> P | Specialized Instruction |
| <input type="checkbox"/> F | Private Lesson | <input type="checkbox"/> Q | Student Teaching |

Fields below if applicable

If course is taught during a part of term in addition to a full 16-week term please indicate the length of the course (ex., 8 weeks): Any format

TCCN (Use for lower division courses) :

Prerequisite(s):		
Course Number/ Placement Test	Minimum Grade Required/ Test Scores	Concurrent Enrollment Permitted? (Y/N)
MATH 1508 or equivalent	C	N

Corequisite Course(s):

Equivalent Course(s):

Restrictions:	
Classification	
Major	LDCS or CS

CS 2101 Discrete Structures I

Course Objectives: *The purpose of this course is to learn to use discrete structures that are central to computer science. In particular, this course provides the foundation for and practice in analyzing computer algorithms concerning both correctness and performance, by introducing logic, proofs, sets, relations, functions, and counting.*

In this class, students are expected to be active learners. They will develop team-working skills, critical-thinking skills, and professionalism.

Prerequisite: *MATH 1508 with a grade of C or better.*

Logistics: **Lecture sessions:** *T 12:00p.m.-1:20p.m. and R 12:00p.m.-12:50 p.m. in CCSB 6-0208*

Instructor: *Dr. Julio Urenda - jcurenda@utep.edu*

- office room: LIB 504

Office hours: *TWR 8:30a.m.-*

9:00a.m. or by appointment

Textbook: *Discrete Math, by Zybooks, available. To subscribe to your textbook, please follow the instructions below:*

- 1. Sign in or create an account at learn.zybooks.com**
- 2. Enter zyBook code: UTEPCS1190Spring2019**
- 3. Subscribe**

Communication platform: *This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.*

Find our class page at: <https://piazza.com/utep/spring2019/cs1190/home>

Grading

Grades are communicated to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of homework assignments, weekly quizzes, class participation, 1 mid-term exam, and a final exam.

The approximate percentages are as follows:

- 30% Homework
- 20% Quizzes
- 45% Exams (1 mid-term exam and 1 final exam)
- 5% Class participation (includes on-time lecture attendance, active participation in class, completion of any quizzes for attendance and survey purposes)

The nominal percentage-score-to-letter-grade conversion for CS 1190 is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- Below 60% is an F

Note: Regardless of your standing in the class at that time, you need to earn a 65 or better at the final exam to pass the course.

Expectations

Class Participation: Attendance at and participation in all lecture sessions are critical factors of your success in this course.

Students should be on time for all scheduled sessions and **attend the entire session**. Attendance will be taken at every session (at first you will have to sign in but as time goes the instructor will know you and mark you present without your help) and will count towards your class participation grade.

Students should notify the instructor prior to missing a session if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence).

It is the student's responsibility to obtain the content covered during missed class(es). Participation points also include completing post-lecture and post-labs online quizzes (when

requested) that are administered as surveys to monitor students' overall progress and potential struggles.

Quizzes: The purpose of each quiz is to ensure that students are staying current with the weekly reading and homework assignments, and to verify that they have acquired the skills developed in class. Quizzes will be administered approximately once a week. There will be **no make-up** on missed quizzes.

Homework: Reading and homework assignments will be announced in class and/or posted on piazza (under the Homework section of Resources). If you miss a lecture session, it is your responsibility to find out what you missed. You should expect to spend **at least two hours per week outside of lecture** on reading and homework assignments and reviews. Most of your homework will be work assigned on your online zybook: completing the assigned activities on time will be crucial to your success in the class (since these activities prepare you for classwork) and to getting a good grade (since late completion will be penalized).

Exams: There will be **one midterm exam** and **one final exam**. These exams together will weigh 50% of your overall final grade for CS1190. Because the exams contribute so heavily to your total grade, it is vital that you do well on them. If you have test-taking difficulties in general or if you have difficulties with our tests in particular, please come and let me know as soon as possible and/or request appropriate accommodation from UTEP's Center for Accommodation and Students' Services.

The purpose of the midterm exam is to allow you to demonstrate mastery of course concepts covered thus far during the semester. The mid-term exam will take place during the regular lecture session and is tentatively scheduled around half of the minimester. Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, **BEFORE** the exam.

The final exam will be comprehensive. You must score 65% or better on the final exam to pass this course. If you have a scheduling conflict (e.g., if you are taking a final at EPCC) or if you are scheduled for three final exams in one day, see your instructor in advance for accommodation, **before the end of the minimester**. The final exam schedule is available online. It is the students' responsibility to keep informed.

Resources

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. CASS' staff are the only individuals who can validate and if need be, authorize accommodations for students with disabilities.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, and submission for credit of any work or materials that are attributable to another person.

Cheating is: Copying from the test paper of another student
Communicating with another student during a test to be taken individually
Giving or seeking aid from another student during a test to be taken individually
Possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.)
Substituting for another person to take a test
Falsifying research data, reports, academic work offered for credit

Plagiarism is: Using someone's work in your assignments without the proper citations
Submitting the same paper or assignment from a different course, without direct permission of instructors
To avoid plagiarism, see: <http://sa.utep.edu/osccr/wp-content/uploads/sites/8/2012/09/Avoiding-Plagiarism.pdf>

Collusion is: Unauthorized collaboration with another person in preparing academic assignments

Important! When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure.

Detailed Learning Outcomes

Level 1: Knowledge and Comprehension. Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. Upon successful completion of this course, students will be able to describe, at a high level the use of:

1. Counting and recurrence relations in computer science to identify time complexity and to analyze its effects on the performance of computer programs, including notions of computability.
2. Inductively defined sets and induction to formulate recursive definitions, abstract data types, and programs.
- 2.3. Predicate logic in program correctness and loop invariants.

Level 2: Application and Analysis. Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

1. Apply basic logical reasoning in propositional logic (e.g., resolution, tableaux) as fundamental components of inference engines in declarative programming, relational databases, and rule-based systems.
2. Utilize the notion and properties of sets, functions, and relations (along with their relation to logic), e.g., in the definition of method signatures, the choice of abstract data types, and the architecture of inheritance between types.

Level 3 Outcomes: Synthesis and Evaluation. Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to:

1. Problem solving mechanics: what does it mean to solve a constraint problem, an optimization problem, an existential / universal problem? What are the common problem solving techniques, including divide and conquer (and how it is based on inductively defined sets and proofs by induction), proof by contradiction and contrapositive proofs (and how they relate to logical reasoning), and design of counter examples.
2. Elements of software testing, such as combinatorial testing, code coverage, via the study of propositional and predicate logic: namely, propositions and operators, representation of knowledge in propositional logic, evaluation of propositions, relation to conditional statements, logical equivalence, partitions and equivalence relations.

COURSE ADD

All fields below are required

College : Engineering

Department : Computer Science

Rationale for adding the course:

This course (with prerequisite CS 2101) is an alternate prerequisite of MATH 2300 Discrete Mathematics for CS upper level courses. The material aligns with MATH 2300, but is more focused to the Computer Science student's interests.

All fields below are required

Subject Prefix and # CS 2202

Title (29 characters or fewer): Discrete Structures II

Dept. Administrative Code : 720

[CIP Code](#) 110701

Departmental Approval Required Yes No

Course Level UG GR DR SP

Course will be taught: Face-to-Face Online Hybrid

How many times may the course be taken for credit? (Please indicate 1-9 times): 1

Should the course be exempt from the "Three Repeat Rule?" Yes No

Grading Mode: Standard Pass/Fail Audit

Description (600 characters maximum):

The course focuses on applications of discrete structures that are central to computer science. It provides a strong foundation and practice in analyzing computer algorithms for correctness and performance. Topics include propositional and predicate logic and their relation to program design and validation, proofs, counting and probability, tree and graph algorithms, and finite state machines.

Contact Hours (per week): 2 Lecture Hours Lab Hours Other

Types of Instruction (Schedule Type): Select all that apply

- | | | | |
|---------------------------------------|-------------------|----------------------------|-------------------------------------|
| <input checked="" type="checkbox"/> A | Lecture | <input type="checkbox"/> H | Thesis |
| <input type="checkbox"/> B | Laboratory | <input type="checkbox"/> I | Dissertation |
| <input type="checkbox"/> C | Practicum | <input type="checkbox"/> K | Lecture/Lab Combined |
| <input type="checkbox"/> D | Seminar | <input type="checkbox"/> O | Discussion or Review (Study Skills) |
| <input type="checkbox"/> E | Independent Study | <input type="checkbox"/> P | Specialized Instruction |
| <input type="checkbox"/> F | Private Lesson | <input type="checkbox"/> Q | Student Teaching |

Fields below if applicable

If course is taught during a part of term in addition to a full 16-week term please indicate the length of the course (ex., 8 weeks): Any format

TCCN (Use for lower division courses) :

Prerequisite(s):		
Course Number/ Placement Test	Minimum Grade Required/ Test Scores	Concurrent Enrollment Permitted? (Y/N)
MATH 1411 AND CS 2101	C	N

Corequisite Course(s):

Equivalent Course(s):

Restrictions:	
Classification	
Major	

CS 2202 Discrete Structures II

Course Objectives: The course focuses on applications of discrete structures that are central to computer science. It provides a strong foundation and practice in analyzing computer algorithms for correctness and performance. Topics include propositional and predicate logic and their relation to program design and validation, proofs, counting and probability, tree and graph algorithms, and finite state machines.

In this class, students are expected to be active learners. They will develop team-working skills, critical-thinking skills, and professionalism.

Prerequisite: CS 2101 and MATH 1411 with a grade of C or better. This is only a mathematical maturity requirement.

Logistics:

Lecture sessions: M-R 4:20p.m.- 6:15 p.m. in CCSB 1.0702

Instructor: Dr. Julio Urenda – jcurenda@utep.edu – office room: LIB 504

Office hours: TWR 12:00p.m.-1:00p.m. or by appointment

Textbook: *Discrete Math*, by Zybooks, available. To subscribe to your textbook, please follow the instructions below:

UTEPCS1290Summer2019

Communication platform: This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com.

Find our class page at: <https://piazza.com/utep/spring2019/cs1290/home>

Grading

Grades are communicated to students in a timely manner. It is the students' responsibility to keep track of their grades by compiling the grades they receive. Your semester grade will be based on a combination of homework assignments, weekly quizzes, class participation, 1 mid-term exam, and a final exam.

The approximate percentages are as follows:

- 40% Homework
- 15% Quizzes

- 40% Exams (1 mid-term exam and 1 final exam)
- 5% Attendance and active participation

The nominal percentage-score-to-letter-grade conversion for CS 1290 is as follows:

- 90% or higher is an A
- 80-89% is a B
- 70-79% is a C
- 60-69% is a D
- Below 60% is an F

Note: Regardless of your standing in the class at that time, you need to earn a 65 or better at the final exam to pass the course.

Expectations

Class Participation: Attendance at and participation in all lecture sessions are critical factors of your success in this course.

Students should be **on time** for all scheduled sessions and **attend the entire session**. Attendance will be taken at every session (at first you will have to sign in but as time goes the instructor will know you and mark you present without your help) and will count towards your class participation grade.

Students should **notify the instructor prior to missing a session** if at all possible, and certainly right after if earlier was not possible. The instructor will allow two unexcused absences per semester before having the option to deduct points from the final grade (5 points per subsequent unexcused absence).

It is the student's responsibility to obtain the content covered during missed class(es). Participation points also include completing post-lecture and post-labs online quizzes (when requested) that are administered as surveys to monitor students' overall progress and potential struggles.

Quizzes: The purpose of each **quiz** is to ensure that students are staying current with the weekly reading and homework assignments, and to verify that they have acquired the skills developed in class. **Quizzes will be administered approximately once a week. There will be no make-up on missed quizzes.**

Homework: Reading and homework assignments will be announced in class **and/or posted on piazza (under the Homework section of Resources)**. If you miss a lecture session, it is your responsibility to find out what you missed. You should expect to spend at least two hours per week outside of lecture on reading and homework assignments and reviews. Most of your homework will be work assigned on your online zybook: completing the assigned activities on

time will be crucial to your success in the class (since these activities prepare you for classwork) and to getting a good grade (since late completion will be penalized). Your homework grade will be a combination of the zybook homework and additional assigned work from class.

Exams: There will be one midterm exam and one final exam. These exams together will weigh 50% of your overall final grade for CS1190. Because the exams contribute so heavily to your total grade, it is vital that you do well on them. If you have test-taking difficulties in general or if you have difficulties with our tests in particular, please come and let me know as soon as possible and/or request appropriate accommodation from UTEP's Center for Accommodation and Students' Services.

The purpose of the **midterm exam** is to allow you to demonstrate mastery of course concepts covered thus far during the semester. The mid-term exam will take place during the regular lecture session and is tentatively scheduled around half of the minimester. Make-up exams will be given only in extremely unusual circumstances. If you must miss an exam, please meet with an instructor, BEFORE the exam.

The **final exam** will be comprehensive. You must score 65% or better on the final exam to pass this course. If you have a scheduling conflict (e.g., if you are taking a final at EPCC) or if you are scheduled for three final exams in one day, see your instructor in advance for accommodation, **before the end of the minimester**. The final exam schedule is available online. It is the students' responsibility to keep informed.

Resources

Special Accommodations: If you have a disability and need classroom accommodations, please contact the Center for Accommodations and Support Services (CASS) at 747-5148 or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.sa.utep.edu/cass. CASS' staff are the only individuals who can validate and if need be, authorize accommodations for students with disabilities.

Scholastic Dishonesty: Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes, but not limited to cheating, plagiarism, collusion, and submission for credit of any work or materials that are attributable to another person.

Cheating is: Copying from the test paper of another student
Communicating with another student during a test to be taken individually
Giving or seeking aid from another student during a test to be taken individually
Possession and/or use of unauthorized materials during tests (i.e. crib notes, class notes, books, etc.)
Substituting for another person to take a test
Falsifying research data, reports, academic work offered for credit

Plagiarism is: Using someone's work in your assignments without the proper citations
Submitting the same paper or assignment from a different course, without direct permission of instructors

To avoid plagiarism, see: <http://sa.utep.edu/osccr/wp-content/uploads/sites/8/2012/09/Avoiding-Plagiarism.pdf>

Collusion is: Unauthorized collaboration with another person in preparing academic assignments

Important! When in doubt on any of the above, please contact your instructor to check if you are following authorized procedure.

Detailed Learning Outcomes

Level 1: Knowledge and Comprehension. Level 1 outcomes are those in which the student has been exposed to the terms and concepts at a basic level and can supply basic definitions. On successful completion of this course, students will be able, at a high level, to:

1. Describe why different types of graphs and trees are better suited to solve specific computer science problems concerning algorithm performance and expressiveness of the data structures.

Level 2: Application and Analysis. Level 2 outcomes are those in which the student can apply the material in familiar situations, e.g., can work a problem of familiar structure with minor changes in the details. Upon successful completion of this course, students will be able to:

1. Use counting principles to identify time/space complexity of computing algorithms and, e.g., how permutations (transpositions) allow to analyze the correctness of sorting algorithms.
2. Apply number theory principles, and in particular modular arithmetic, to computer science problems including cryptography.
3. Identify and motivate the choice of proper data structures to implement algorithms (e.g., using finite state machines) to solve an array of computer science problems, e.g., social networks, neural networks, shortest path, call-in shortest path, and data mining.

Level 3 Outcomes: Synthesis and Evaluation. Level 3 outcomes are those in which the student can apply the material in new situations. This is the highest level of mastery. On successful completion of this course, students will be able to:

1. Use counting principles to identify time / space complexity, in particular as it relates to graph/tree-based problems.
2. Lay out a proof plan for existential and universal problems and identify shortcomings of some types of proving strategies.

3. *Identify the inductive structure of a structure (e.g., set, function) and use it to conduct inductive proofs and identify recurrence relations.*
4. *Design proper pre- and post-conditions, as well as proofs using loop invariants based on predicate logic.*