CIT 5920

Mathematical Foundations of Computer Science

Or how to communicate formally

FALL 2025 Jérémie Lumbroso, Ph.D.
August 26th, 2025
University of Pennsylvania
Computer & Information Science

Welcome!

Today's Agenda

- Meet your instructor
- Understand the "why" behind this course
- Course overview and expectations
- Get connected with participation tools
- Key policies and resources

Audience Participation

We'll use Sli.do throughout the semester

- Ask questions and upvote others' questions
- Answer polls and quizzes
- Choose to be anonymous or not

Join now: https://sli.do
Event code: #6771

We'll pause for Sli.do activities throughout class - please participate!

Meet Your Instructor

Jérémie Lumbroso

- Originally from France
- MS, Ph.D from Sorbonne Université
- Previously taught at:
  - Simon Fraser University (Vancouver, 2013-2014)
  - Princeton University (2014-2023)
- Created education startups: codePost, CUvids

My Teaching Philosophy

Your success is my top priority

- We are here for you
- I focus on engagement with the material
- I prioritize your questions
- Grading is secondary to learning
- YOU are the superstars - help each other!

What I admire most:

- Kindness
- Helping others
- Patience

Important: Neurodivergence

I am neurodivergent (HSP autistic), my brain operates differently

Please help me help you:

- **Hypersensitive to touch**: Please don't touch me or my belongings
- **Hypersensitive to noise**: Please be quiet as we begin lectures
- **Hypersensitive to smell**: Please no food in the classroom.
- **Hypersensitive to time pressures**:
  - Please be patient with me
  - Don't come to my office without an appointment

When I am grounded, my goal is always to make you happy and fulfilled.

Teaching Assistants

Head TAs: Dennis (Zhe) Zhang & Tiffany (Yixuan) Gao

What to expect from TAs:

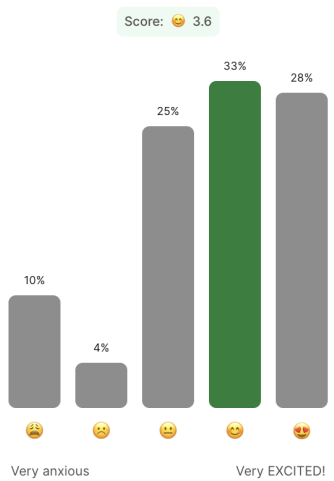
- Hand-recruited experts who care about your success
- They've taken this exact course
- Your most reliable resource
- Office hours throughout the week
- Recitation sessions on Fridays

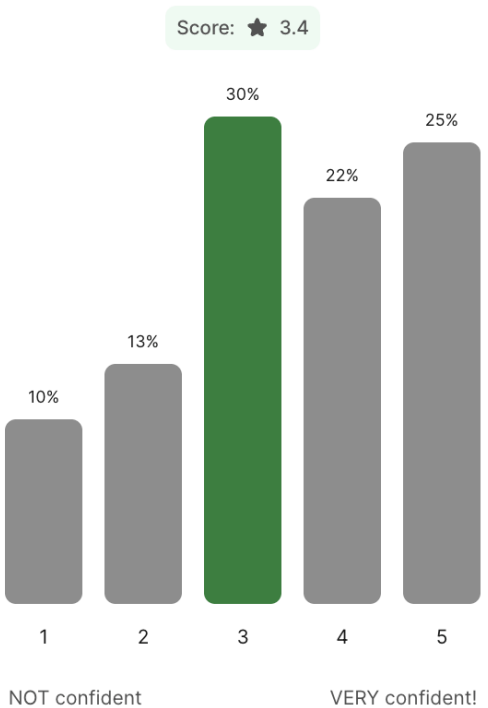
Quick Class Survey

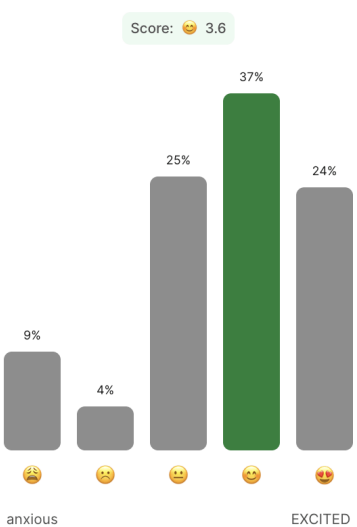
Let's learn about each other!

[POLL: Where do you come from?]
[POLL: What's your undergraduate major?]
[POLL: Current career or job role?]
[POLL: How do you feel about starting MCIT?]
[POLL: Mathematical confidence level?]

Last Year's Survey!







Why We Prioritize Introductions

The Hidden Curriculum: Your Network

The Research Shows:

Collaborative learning increases retention by 40%
Study partners improve grades by a full letter on average

The Reality Check:

Week 1-2: Everyone's open to meeting
Week 3+: Social groups solidify, harder to join
Week 8: "I wish I knew more people in class"

Our Solution:

✓ Multiple structured introduction opportunities
✓ Speed networking in recitation
✓ Facilitated study partner matching
✓ Breaking awkwardness before it settles in

Your degree value = 40% what you know + 60% who you know

Don't wait - your future study group is in this room today

Every time you talk to me (or each other)

Start by introducing yourself: "*Hey I'm X: …*"

Why This Course?

The Big Picture

- Foundation for CIS 5020: Analysis of Algorithms
- Foundation for CIT 5960: Algorithms and Computation
- Develops formal computational thinking
- Enables precise technical communication
- Builds confidence for independent learning

Sample Evaluation

Jeremie Lumbroso--the best math Professor ever! I grew up taking math courses, but he is the only teacher who really made me learn something. He not only taught us math--he taught us the way to think about things, which is far more important than the concepts per se. He was always sincere. Some other math teachers like to impress: they learned things by heart themselves, and they talked as if they know everything since they were born. Jeremie is different. He paused to ponder over things, and he pointed out the ambiguity and the questions people may have. He is able to be vulnerable himself, which makes him super strong. I must say I got more than I expected from this course. Love you Jeremie! I'm sure 5920 will be popular since now because of you.

QuickStudy

Sets

Definitions

A **set** is a collection of objects (such as numbers). Each member is called an **element** of the set. If A and B are sets, A is a **subset** of B if and only if every element of A is also an element of B. This is written $A \subseteq B$ which is read "A is a subset of B" or "B contains A."

If C and D are sets, C is a **proper subset** of D if and only if every element of C is an element of D, but there is at least one element of D that is not in C.

EX:

- The set N = {red, blue, green, yellow, purple, orange} has 6 elements.
- The set M = {red, green, yellow, orange} is a proper subset of N. Every element of M is an element of N, but there is at least one element in N that is not in M.

A **Venn diagram** is a visual representation that shows the relationship between sets.

EX: The set of whole numbers $\mathbf{W}$ = {0, 1, 2, 3, ...} is a proper subset of the set of integers $\mathbf{Z}$ = {..., −3, −2, −1, 0, 1, 2, 3, ...}.

Unions, Intersections & Complements

In mathematics, a **universal set**, U, is the set of all elements under consideration for a particular situation. The universal set could be the set of all integers, the set of all real numbers, the set of all positive integers, etc.

For a universal set U and sets A and B with $A \subseteq U$ and $B \subseteq U$, the following definitions hold.

- The **union** of A and B is the set of all elements x in U such that $x \in A$ or $x \in B$. The notation is $A \cup B$.
- The **intersection** of A and B is the set of all elements x in U such that $x \in A$ and $x \in B$. The notation is $A \cap B$.
- The **complement** of A is the set of all elements x in U such that $x \notin A$. The notation is A'.

EX: Let U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}, R = {2, 4, 6, 8, 10}, and S = {1, 2, 3, 4, 5}.

$R \cup S$ = {1, 2, 3, 4, 5, 6, 8, 10}

$R \cap S$ = {2, 4}

$R' = \{1, 3, 5, 7, 9\}$

$S' = \{6, 7, 8, 9, 10\}$

Properties of Sets

For subsets A, B, and C of a universal set U, the following identities are true.

- **Associative Property**
  $(A \cap B) \cap C = A \cap (B \cap C)$
  $(A \cup B) \cup C = A \cup (B \cup C)$
- **Commutative Property**
  $A \cap B = B \cap A$
  $A \cup B = B \cup A$
- **Distributive Property**
  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
  $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- **Complement of a Complement Property**
  $(A')' = A$
- **De Morgan's Laws for Sets**
  $(A \cap B)' = A' \cup B'$
  $(A \cup B)' = A' \cap B'$

Ordered n-tuples & Cartesian Products

An **ordered n-tuple** is an arrangement of n elements of a set in which the order is important. The notation for an ordered n-tuple is $(x_1, x_2, \ldots, x_n)$ in which x_1 is first, x_2 is second, and so on. An ordered 2-tuple is also called an **ordered pair** (x, y), and an ordered 3-tuple is called an **ordered triple** (x, y, z).

Two ordered n-tuples $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$ are equal if and only if $x_1 = y_1, x_2 = y_2, \ldots, x_n = y_n$.

For two sets A and B, the **Cartesian product** of A and B is the set of all ordered pairs (a, b), where $a \in A$ and $b \in B$. The notation for a Cartesian product is $A \times B$ which is read "A cross B."

EX: Given the sets A = {1, 2} and B = {3, 4, 5}, find A × B.

A × B = {(1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5)}

The Empty Set

The **empty set** (or null set) is a unique set that contains no elements. The empty set is represented symbolically as Ø or {}. There are several properties of sets that involve the empty set. For a universal set U and all sets A, the following properties are true:

- $A \cup \varnothing = A$
- $A \cap A' = \varnothing$
- $A \cup A' = U$
- $A \cap \varnothing = \varnothing$
- $U' = \varnothing$
- $\varnothing' = U$

Partitions of Sets

Two sets are **disjoint** if the sets have no elements in common.

A **partition** of a set A is a collection of nonempty sets $\{A_1, A_2, \ldots, A_n\}$ such that $A = A_1 \cup A_2 \cup \ldots \cup A_n$ and $A_1, A_2, \ldots, A_n$ are mutually disjoint.

EX: Let A = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. A partition of set A is $\{A_1, A_2, A_3\}$, where A_1 = {1, 2}, A_2 = {3, 4, 5, 6, 7}, and A_3 = {8, 9, 10}.

Counting

An experiment or process is **random** if it is not possible to predict the outcome with certainty for each trial. It is often useful to be able to count all of the possible outcomes of a random process or the subsets of a given set.

The Fundamental Counting Principle

The **Fundamental Counting Principle** says that if a process A has a possible outcomes, process B has b possible outcomes, process C has c possible outcomes, . . . , and process N has n possible outcomes, then the total number of ways all of the processes can occur is $a \cdot b \cdot c \cdot \ldots \cdot n$.

EX: A deli offers 5 different kinds of bread, 8 different kinds of sandwiches, and 7 different side dishes. How many possible lunches are there if a customer selects a kind of bread, type of sandwich, and side dish at random?

There are $5 \cdot 8 \cdot 7 = 280$ possible lunches.

Combinations

A **combination** is a set of objects in which the order of the objects is not important.

EX: There are 12 books on a shelf. Meredith selects 2 of the books at random to take on vacation. How many possible combinations of books are there?

$n = 12, r = 2 \quad {}_{12}C_2 = \dfrac{12!}{(12-2)!2!} = \dfrac{12!}{10!2!} = 66$

Permutations

A **permutation** is a set of objects in which the order of the objects is important. A set that contains n elements has $n!$ different permutations. This is because there are n possible outcomes for the first choice, $(n - 1)$ possible outcomes for the second choice, and so on.

EX: How many permutations are there of the elements of the set {red, blue, green}?

There are $3! = 6$ permutations as shown below:

red, blue, green
red, green, blue
blue, red, green
blue, green, red

Counting & Probability

A **sample space** is the set of all possible outcomes for a random experiment or process. An **event** is a subset of a sample space of outcomes. The **probability** of an event is a numerical value that describes how likely the event is to occur. Probability ranges from 0 (impossible) to 1 (certain). It is calculated as the ratio of the number of favorable outcomes to the total number of possible outcomes in the sample space. The probability of an event E is denoted $P(E)$. The counting methods described earlier play an important role in calculating probability.

$$P(\text{event}) = \frac{\text{number of favorable outcomes}}{\text{total number of outcomes}}$$

EX: There are 6 sophomores, 8 juniors, and 10 seniors on a committee. A subcommittee of 6 students is chosen at random. What is the probability that the subcommittee contains 2 sophomores, 2 juniors, and 2 seniors?

Find the total number of outcomes (combinations)

QuickStudy

Functions & Set Relations

Definitions

A **relation** R from set A to set B is any subset of the Cartesian product $A \times B$. For a given ordered pair (x, y) in $A \times B$, x R y (read "x is related to y by R") if and only if $(x, y) \in R$.

EX: Let A = {1, 2, 3} and B = {4, 5}. Define the relation R to be any {x, y} in A × B such that $|y - x|$ is odd. Determine which ordered pairs are in R.

A × B = {(1, 4), (1, 5), (2, 4), (2, 5), (3, 4), (3, 5)}

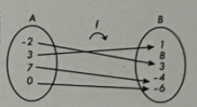
Of these ordered pairs, only (1, 4), (2, 5), and (3, 4) satisfy the rule $|y - x|$ is odd.

So, R = {(1, 4), (2, 5), (3, 4)}.

A **function** f from set A to set B is a relation between the elements of A and the elements of B such that each element of A is related to a unique element of B. The notation $f: A \rightarrow B$ is read "f is a function from A to B."

A is the **domain**, or the set of possible input values for the function. B is the **codomain**, or the set of possible output values for the function. The **range** of a function is the actual set of output values from the codomain that are paired with input values from the domain. A **mapping diagram** uses arrows to show how elements from the domain are paired with elements from the codomain.

EX: The mapping diagram shows a function f: $A \rightarrow B$. Identify the domain, codomain, and range of the function. Then list the ordered pairs in f.



The domain is the set of possible input values, or {−2, 3, 7, 0}.

The codomain is the set of possible output values, or {1, 8, 3, −4, −6}.

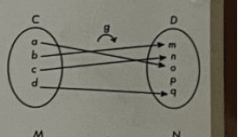
The range is the set of output values that are paired with input values from the domain, or {1, 3, −6}. f = {(−2, 3), (3, 1), (7, −6), (0, −6)}

Injective, Surjective & Bijective Functions

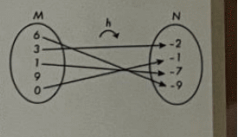
A function $f: A \rightarrow B$ is **injective**, or **one-to-one**, if and only if each element of the codomain is mapped by *at most* one element of the domain. In other words, if x_1 and x_2 and elements of the domain where $x_1 \neq x_2$, then $f(x_1) \neq f(x_2)$.

EX: The function g: $C \rightarrow D$ shown in the mapping diagram is injective because no two elements of the domain are mapped to the same element of the codomain.

A function $f: A \rightarrow B$ is **surjective**, or **onto**, if and only if for each element of the codomain there is at least one element in the domain that maps to it. In other words, if y_1 is any element of the codomain B, then there exists an element x_1 in A such that $f(x_1) = y_1$. For a surjective function, the range is equivalent to the codomain.



EX: The function h: $M \rightarrow N$ shown in the mapping diagram is surjective because each element of the codomain has an element of the domain that maps to it.

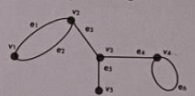


A function $f: A \rightarrow B$ that is both injective and surjective is a **bijective function**. Another way of saying that a function is bijective is to say that there is a **one-to-one correspondence** from A to B.

Graphs & Trees

Graphs

A **graph** is defined by a finite set of points, called **vertices**, and a finite set of line segments, called **edges**, that connect the vertices. Each edge is associated with either one or two vertices. An edge that is associated with just one vertex is called a **loop**. Two distinct edges that are associated with the same set of vertices are **parallel** edges. A **simple graph** is a graph that does not have any loops or parallel edges.



EX: Describe the edges and vertices of the graph.

- The vertices of the graph are $\{v_1, v_2, v_3, v_4, v_5\}$.
- The edges of the graph are $\{e_1, e_2, e_3, e_4, e_5, e_6\}$.
- Edges e_1 and e_2 are parallel because they are distinct edges that are both associated with vertices v_1 and v_2.
- Edge e_6 is a loop because it is associated with only one vertex, v_4.

Walks, Paths & Circuits

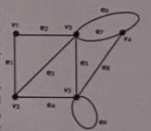
For a given graph with vertices v_1 and v_2, a **walk** from v_1 to v_2 is an alternating sequence of adjacent vertices and edges.

EX: In the preceding graph, one walk from v_1 to v_3 would be $v_1e_2v_2e_3v_3e_4v_4e_5v_5$.

A **path** from v_1 to v_2 is a walk that does not contain any repeated edges. A **simple path** is a path that does not contain any repeated vertices. The walk described in the previous example is a simple path because none of the edges or vertices in the walk repeat.

A **closed walk** is a walk that begins and ends at the same vertex. A **circuit** is a closed walk that does not contain a repeated edge. A **simple circuit** is a circuit with no repeated vertices except for the beginning and ending vertex.

EX: For this graph shown, describe each of the following walks as a path, a simple path, a circuit, or a simple circuit.

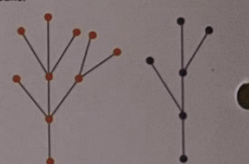


- $v_5e_6v_4e_4v_3e_3v_2e_1v_1$ is a simple path. It is a walk from v_5 to v_1 and has no repeated vertices or edges.
- $v_1e_2v_2e_4v_4e_3v_3e_3v_2e_1v_1$ is a circuit. It is a closed walk from v_1 to v_1 with no repeated edges, but it does have repeated vertices (v_3).
- $v_1e_2v_2e_4v_4e_3v_3e_5v_5$ is a path. It is a walk from v_1 to v_5 with no repeated edges, but it does have repeated vertices (v_3).
- $v_2e_3v_3e_4v_4e_5v_5$ is a simple circuit. It is a closed walk from v_2 to v_2 with no repeated edges or vertices other than the beginning and ending vertex.

Trees

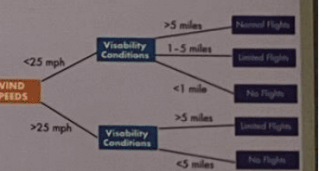
Two vertices of a graph, v_1 and v_2, are **connected** if and only if there is a walk from v_1 to v_2. A graph is connected if and only if every pair of vertices in the graph is connected. A **trivial circuit** is a circuit that consists of only a single vertex. A graph is a **tree** if and only if it is connected and has only trivial circuits.

EX: The graphs shown below are trees. Every pair of vertices is connected and the only circuits are trivial circuits.



Decision trees are used in a variety of situations to make decisions based on a number of factors.

EX: The decision tree below could be used to determine whether a hot air balloon company flies depending on the wind speeds and visibility conditions.



U.S. $6.95

What We'll Cover

Part 1: The Characters (Objects)

- Sets, functions, relations
- Permutations, combinations
- Probability and expectations

Part 2: The Stories (Proofs)

- Propositional logic
- Proof methods
- Mathematical induction

Part 3: The Wilderness (Graphs)

- Introduction to graph theory
- Applications in CS

Real-World Applications: Sets

Database Operations

```sql
SELECT EMP_ID, LAST_NAME
FROM EMPLOYEE_TBL
WHERE LAST_NAME = 'Smith';
```

Sets are fundamental to:

- Database queries
- Data filtering
- Collection operations

Real-World Applications: Logic

Simplifying Code & Digital Circuits

Logical reasoning helps:

- Simplify complex conditional statements
- Design digital circuits
- Debug programs systematically

Example: All MCIT students → Looking for new jobs
You're in MCIT → We'll help with job search!

Real-World Applications: Proofs

Writing Bug-Free Code

The ability to write proofs ≈ Writing programs with few bugs

- Both require translating thoughts into unambiguous expressions
- Essential for critical systems (medical devices, aviation)
- Foundation for algorithm analysis

"Beware of bugs in the above code; I have only proved it correct, not tried it." – Donald Knuth

Real-World Applications: Discrete Math

Penn Engineering
University of Pennsylvania

Makes You:

Clearer

- Abstract problems with notation
- Generalization and parameterization

Faster

- Recursive definitions
- Efficient algorithms

More Correct

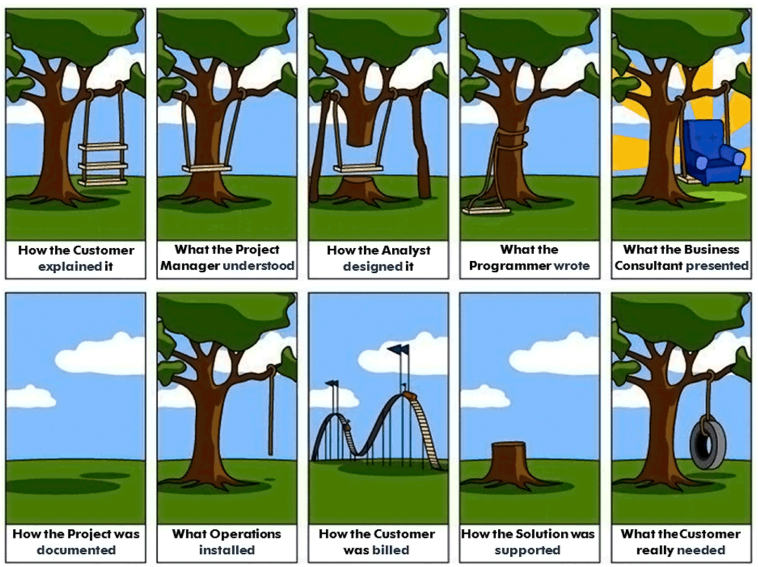
- Formal definitions prevent edge cases
- Proofs ensure correctness

The Power of Mathematical Formalism 1/2

The $10,000 Bug: "Sort the List"

Natural Language Specification:

"Sort the customer records by date"



What Actually Happened:

- Developer A: Sorted by date created
- Developer B: Sorted by date modified
- Developer C: Sorted by date of birth
- Developer D: Sorted newest → oldest
- Developer E: Sorted oldest → newest

Result: Five different implementations, angry customers, weeks of rework

Mathematical Specification:

Given array A[1..n] of records with field A[i].timestamp, produce array B[1..n] such that:

- $\forall i \in \{1..n\}$: $B[i] \in A$
- $\forall i,j$ where $1 \leq i < j \leq n$: $B[i].timestamp \leq B[j].timestamp$
- B is a permutation of A

Result: One interpretation, zero ambiguity

The Power of Mathematical Formalism 2/2

"The System Should Be Fast"

Ambiguous Requirements:

- "Fast response time" → Fast for whom? How fast?
- "Handle many users" → How many is many?
- "Usually works" → What percentage is "usually"?

Formal Specification:

- Response time: 95th percentile < 200ms
- Throughput: ≥ 10,000 requests/second
- Availability: ≥ 99.99% uptime

Even Better - Big-O Notation:

Instead of: *"The search algorithm should be efficient"*
We write: **Time complexity: O(log n)**
Now everyone knows:

- Doubling data size adds just ~1 operation
- 1 billion items need only ~30 operations
- We can make precise architectural decisions

Real-World Applications: Counting & Probability

Where It's Used:

- **Performance Analysis**: How long will my program take?
- **Scalability**: How does runtime grow with data?
- **Machine Learning**: Core of ML algorithms
- **Randomized Algorithms**: Expected runtime analysis
- **Financial Systems**: Risk assessment

Real-World Applications: Graph Theory

Powers Modern Technology:

- **Google**: PageRank algorithm
- **GPS/Maps**: Shortest path algorithms
- **Social Networks**: Friend recommendations
- **Internet Routing**: Network optimization
- **Recommendation Systems**: User-item graphs

Course Resources

Primary Resources

- **Course Website**: https://math4cs.university
- **Discussion Forum**: Ed Discussion
- **Homework Platform**: PrairieLearn
- **LaTeX/Writing**: Overleaf (free Penn account)

Textbooks (Free!)

- Online lecture notes by Prof. Arvind Bhusnurmath
- "A Cool Brisk Walk Through Discrete Mathematics" by Stephen Davies
- Additional free resources on course website

How We'll Learn

Active Learning Approach

Lectures: Core content + Sli.do engagement
Homework: PrairieLearn (practice) + Written proofs (analysis)
Recitations: Hands-on problem solving
Office Hours: Individual help from entire team

Key: Multiple attempts allowed on PrairieLearn!

Grading Philosophy

Distribution

- Weekly Homeworks: 45%
- Three Exams: 45%
- Professionalism: 5%
- Engagement: 5%

Exam Weighting

Best exam: 17%, Second: 15%, Third: 13%

Bottom line: We minimize grading stress so you can focus on learning

Key Policies Overview

Collaboration

- Work with ONE partner allowed
- Must write your own solutions
- Can change partners

AI Usage

- Allowed as an assistant
- **Recommend/urge getting a paid subscription**
- You must understand everything you submit

Extensions

- "No Time Pressures" policy
- Request before deadline
- We'll accommodate your needs

More on AI Usage

General Rules

1. Get a paid subscription
   - No, DeepSeek is *excellent* but not enough
   - Pay for ChatGPT or Claude
2. Learn to differentiate between "learning" and "unlearning"
   - Really important to understand that "pleasing the professor" is not at all important 🤣
3. Think critically always
   - Engage in conversations with agents — don't just ask them to do work, ask them why they do it this way, request improvements, opposing ideas, etc.

Some Practical Tips (to begin!)

- Use transcripts of lectures
- Generate examples, analogies, metaphors
- Combine transcript and lecture notes sources
- Use dictation with Superwhisper

Academic Integrity

This is YOUR education

- Your money, time, and opportunity
- University diploma = more than paper
  - The corrupt viewpoint
- MCIT students are mature learners
- Waiver exam (on website): **If you are not interested**

Critical Rule

Do. Not. Share. Course. Solutions.

Sharing solutions hurts everyone:

- Forces instructors to recreate materials
- Reduces time for improvements
- Undermines education quality

Next Steps

This Week:

1. Join Ed Discussion forum
2. Set up Overleaf account
3. Review course website
4. Attend Friday's special recitation:
    - Speed networking with peers
    - LaTeX tutorial

Homework 1

- Released soon

Friday's Special Recitation

Two Important Activities:

1. Speed Networking

- Meet your classmates
- Find study partners
- Build your support network

2. LaTeX Introduction

- Set up Overleaf
- Learn basic mathematical notation
- Practice for homework submissions

Don't miss it!

Thank You!

Questions?

Remember:

- We're here for your success
- Office hours start next week
- Use Ed Discussion for questions
- Feedback is always welcome

Let's have a great semester!