# CSE215: Lecture 1
# Foundations of Computer Science

Instructor: Zhoulai Fu

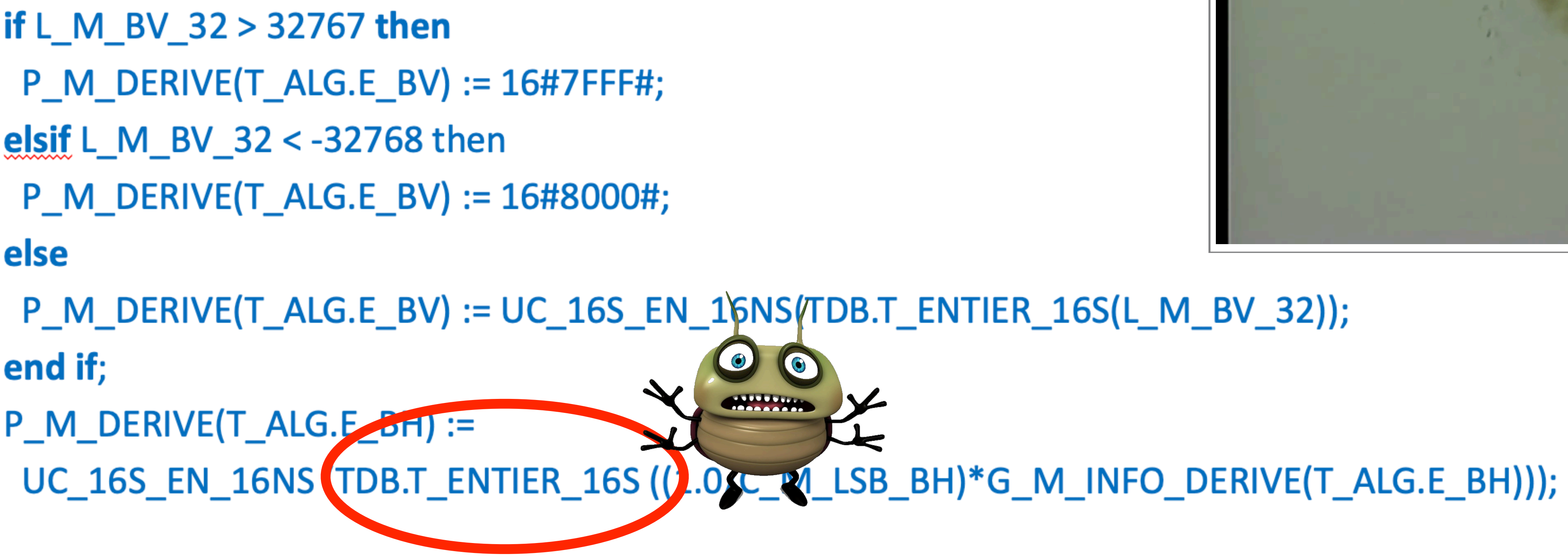
State University of New York, Korea

August 30, 2022

Course materials and Info available here:
https://github.com/zhoulaifu/22_cse215_fall

# French Ariane 5 Rocket, 1996

# Ada code for Ariane 5 Rocket



```
if L_M_BV_32 > 32767 then
  P_M_DERIVE(T_ALG.E_BV) := 16#7FFF#;
elsif L_M_BV_32 < -32768 then
  P_M_DERIVE(T_ALG.E_BV) := 16#8000#;
else
  P_M_DERIVE(T_ALG.E_BV) := UC_16S_EN_16NS(TDB.T_ENTIER_16S(L_M_BV_32));
end if;
P_M_DERIVE(T_ALG.E_BH) :=
  UC_16S_EN_16NS (TDB.T_ENTIER_16S ((1.0/C_M_LSB_BH)*G_M_INFO_DERIVE(T_ALG.E_BH)));
```

## $7 billion Software Disaster

Comparison:

SUNY Korea was awarded $0.05 billion for 10 years under an MKE grant
(Source: https://sunyk.cs.stonybrook.edu/)

From 2018 to 2020, South Korea GDP dropped $94 billion;
(Source: World bank)

Propositional Logic

Predicate Logic

Proof

# Why does a computing system fail (or work)?

Sequences

Sets

Functions

Relations

# Expected Learning Outcomes

- An ability to check if a mathematical argument is valid and sound
- An ability to verify the correctness of proofs of some existing theorems and prove some new theorems
- An ability to use the mathematical concepts of sequences, functions, relations, and sets in solving computing problems

# Meet the Instructor

## Education

- B.Sc, M.Sc, Ecole Polytechnique, France
- M.Eng. Telecom Paris, France
- Ph.D. INRIA (National CS Lab), France

# Teaching & Research

- University of California Davis, United States
- IT University of Copenhagen, Denmark
- SUNY Korea

# Meet the TA

# Each team member’s jobs

| You | TA | Instructor |
|---|---|---|
| Lectures<br>Homework<br>Ask questions | Office hours<br>Answer questions | Office hours<br>Lectures<br>Grading<br>Answer questions |

# Practical matters

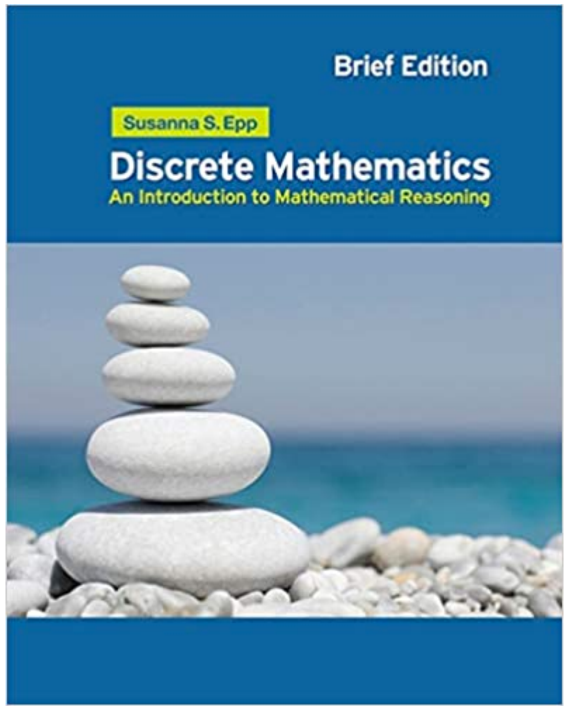
- COVID
- Textbook
- Schedule
- Homework
- Exams and grading
- Ask for help

# Covid

- In-person classes; Indoor masks; Social distance
- Inform instructor immediately of the date of a positive test.
- Follow government guidelines including a 7-day quarantine.
- Return to the class after quarantine. Negative test not needed.

- Let’s do our best to keep us safe
- While maintaining the quality of your learning experience

# Textbook

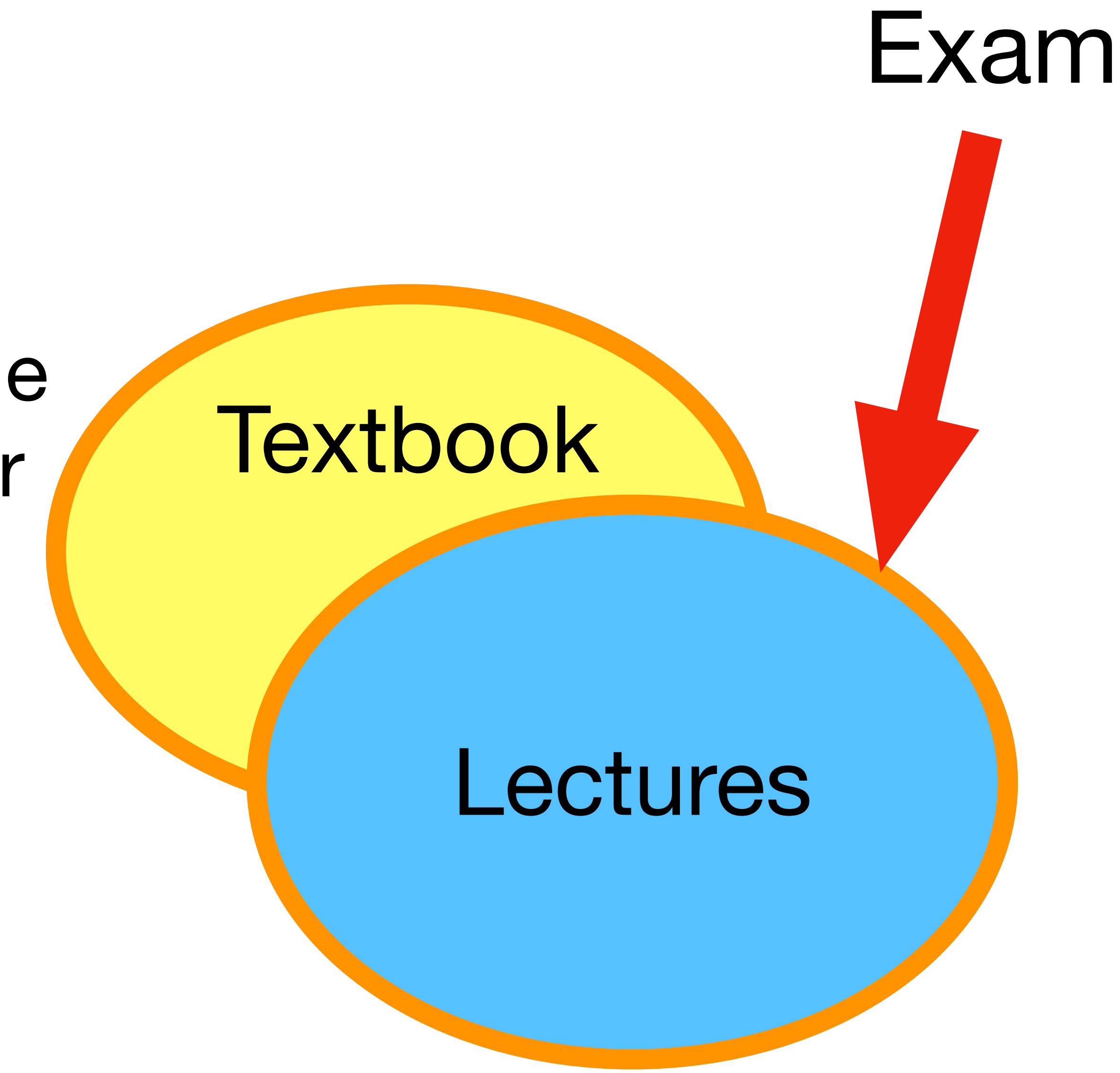


- Our course relates to Chapters 2-7

- Very helpful, though optional

- Suggestion: Skim the related chapter before the lecture; read deeper after the lecture

- Textbook may not cover everything in the exams; lectures do



Exam

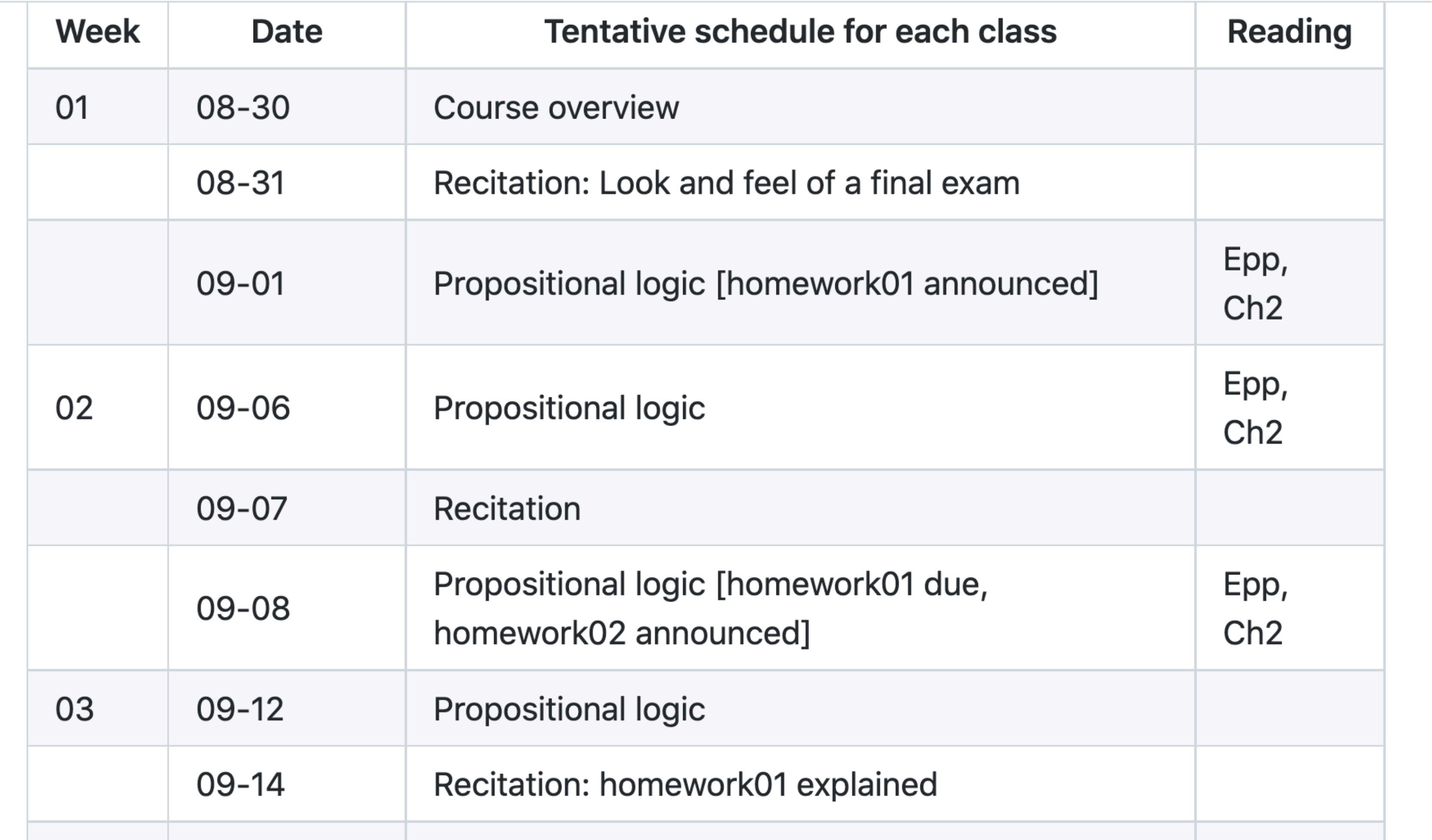
Textbook

Lectures

# Schedule

- Main lectures: Tuesday and Thursday 12h30 pm- 1h50 pm, at B204
- Recitation: Wednesday 3h30pm - 4h25pm, at B203
- Office hours: Tuesday and Thursday 2h15pm - 3h15pm, at B424
- TA office hours: TBA
- Per-class schedule: https://github.com/zhoulaifu/22_cse215_fall#schedule

# Per-class schedule



| Week | Date | Tentative schedule for each class | Reading |
|---|---|---|---|
| 01 | 08-30 | Course overview | |
| | 08-31 | Recitation: Look and feel of a final exam | |
| | 09-01 | Propositional logic [homework01 announced] | Epp, Ch2 |
| 02 | 09-06 | Propositional logic | Epp, Ch2 |
| | 09-07 | Recitation | |
| | 09-08 | Propositional logic [homework01 due, homework02 announced] | Epp, Ch2 |
| 03 | 09-12 | Propositional logic | |
| | 09-14 | Recitation: homework01 explained | |

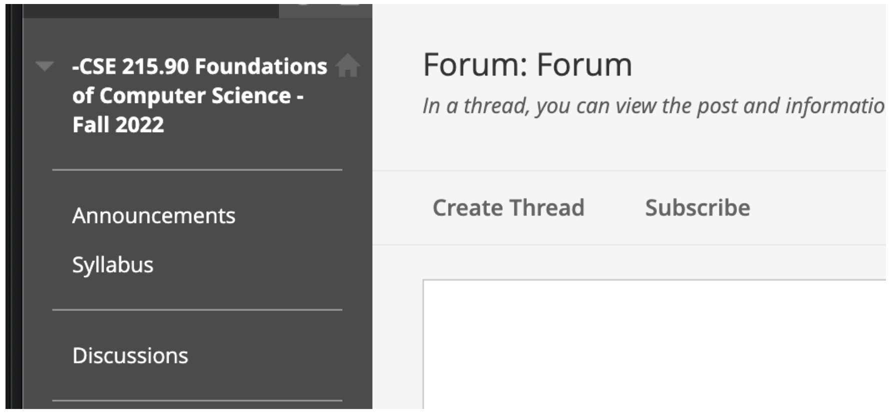
- Slides will be available before lectures start
- Homework: Due. Announced. Explained.
- Schedule may change. Major changes will be announced at Blackboard.

# Exams and grading

- Attendance: 5%
- Homeworks: 45%
- Midterms: 30%
- Final exam: 20%
- Numerical Grading is a sum of
    - total attendance / total check * 100 * 5%
    - (homework01 + … + homework13 ) / 11 * 45%
    - (midterm1 + midterm2) / 2 * 30%
    - Final * 20%

# Questions are gold

- The recommended way is to send questions to Forum at blackboard. You could get answers faster, and benefit others. **Please subscribe** to Forum get email notification.



-CSE 215.90 Foundations of Computer Science - Fall 2022

Forum: Forum

*In a thread, you can view the post and informatio*

- If you are shy, send questions to TA or me

- For urgency, feel free to call the instructor : 010-____- ____

## Questions so far?

# Technic overview

*A personal story*

# The story

- Once upon a time, I worked for a project involving financial calculation
- I needed to sum up a number of floating-point values like
  - 0.1 + 0.2 + 0.3 + 0.7 + 0.9 + 1.2 + 3.5…
- There were billion of numbers like this, so performance was a key for the project's success
- We decided to use the state-of-the-art multi-core, parallel computing
- Parallel computing works like a divide-and-conquer:
  - (0.1 + 0.2) + (0.3 + 0.7) + (0.9 + 1.2 + 3.5) + …
- Now, let us think why it looks reasonable le to use parallel computing for this task??
- The reason is associative law. (a + b) + c = a + (b + c)

# A problem

We get different results for each round, if we put parentheses differently each time.

- (0.1 + 0.2) + (0.3 + 0.7) + (0.9 + 1.2 + 3.5) + …
- becomes different from
- (0.1 + 0.2 + 0.3)  + (0.7+ 0.9 )+ (1.2 + 3.5) + …

**Live demo**

# Why?

- We made this assumption:
  - for any numbers a, b, c, (a + b) + c = a + (b+ c)
- This is a statement that can be assigned with true or false value, we call it a **proposition**
- The inner part has variables, and can be denoted as a statement with parameters (a, b, c). We call it a **predicate.**
- Many CS work involves determining if a proposition is true or false. To show the truth is called **to prove**.
- The reason for the problem is that the proposition above is false.

**Live demo:**

**First check 1 + 2 + 3, then 0.1 + 0.2 + 0.3**

# Summary for the story

The whole is called a proposition,
to which we can assign a truth value

$$\forall a, b, c \in I, \underbrace{(a+b)+c = a+(b+c)}_{\text{Predicate}}$$

quantifier

variables

set

# Exercise

- Assume associative law:
  $\forall a, b, c \in I, (a+b)+c = a+(b+c)$

- Assume 1 + 1 = 2,  2 + 1 = 3

- Try to prove 1 + 2 = 3

# Takeaway for today

- The ultimate goal of this course is to learn fundamentals for understanding why our digital world works or fails.

- We will study logic (propositions, and predicates), proof, and math structures like sets

- Practical matter: covid protocol, schedule, ask questions