CSE215: Foundations of Computer Science

Spring 2020

Stony Brook University

Instructor: Y. Annie Liu

http://www.cs.stonybrook.edu/~liu/cse215

Course Description

- "Introduction to the logical and mathematical foundations of computer science. Topics include functions, relations, and sets; recursion; elementary logic; and mathematical induction and other proof techniques."
- This is NOT a course in computer programming, BUT on fundamental concepts of computing.
- We will stress **mathematical** problem solving skills and the use of **formal** concepts as tools for computer science.
- Prerequisites: AMS 151 or MAT 125 or MAT 131.

General Information

- Meeting information:
  - Lecture section 2: Mondays and Fridays 1 - 2:20 PM, Engineering 145.
  - Recitation section 7: Mon 11-11:53 AM, CS 2129.
  - Recitation section 4: Wed 11-11:53 AM, CS 2129.
  - Recitation section 10: Fri 10-10:53 AM, CS 2114.
  - Recitation section 6: Fri 11-11:53 AM, CS 2114.
- During recitations, the TA will reinforce lecture materials and guide problem solving sessions.

General Information

- Course Web page: http://www.cs.stonybrook.edu/~liu/cse215
- Google Classroom will be used for assignments, grades, and course materials, including recitation sections and Q&A forum.
- Q&A forum should be used for all questions related to this course except for personal issues.

Instructor Information

- Annie Liu
  New Computer Science Building, Room 237
- Office hours: TBD, an online poll in a day
- I am also available by appointment
- Email: liu@cs.stonybrook.edu
  - Please include "CSE 215" in the email subject and your name in your email correspondence

Textbook

- Discrete Mathematics:

  Introduction to Mathematical Reasoning

  Author: Susanna S. Epp

  Publisher: Brooks/Cole Cengage Learning

  Brief edition, 1st edition (2011)

  ISBN-10: 0495826170

  ISBN-13: 978-0495826170

What is Computer Science?

- Why do we study mathematics and problem solving in a major course in Computer Science?
  - Computer Science is NOT computer programming - although programming is part of it.
  - Computer Science is a **mathematical science** - we study the capabilities and limitations of computers and how people can use them effectively.
  - Computer programming requires that the exact specifications to perform a task be specified completely and precisely
    - difficult and requires careful reasoning about **abstract entities**
  - **Mathematics has developed over thousands of years as a method of abstract reasoning.**

Why Isn't CS "Just Programming"?

- Programs of only a few hundred lines are easy for one person to build with little training.
- BUT:
  - Real-world software systems are **large**
    - Developing and understanding such complicated objects requires mental and mathematical discipline.
  - Real-world software systems must be **reliable**
    - They control economies, airplanes, nuclear weapons, and your car.
    - **Systematic** discipline is necessary to avoid errors
- Mathematics provides the disciplined and systematic language to reason about such systems.

Important dates

- Midterm exam 1: Fri 3/06/2020, 1- 2:20 PM, Engineering 145.
- Midterm exam 2: Fri 4/17/2020, 1 - 2:20 PM, Engineering 145.
- Final exam: Monday, May 18, 2020, 2:15 - 5 PM, Room TBD.
- The exams will be like what we solve in the class!

Course work

- **Grading**
  - Lecture critique: 2%
  - In-class exercises: 8%
  - Homework assignments: 20%
  - Midterm exams: 40% (20% each)
  - Final exam:30%

Lecture critique

- **Each student critiques one lecture**
  - It is worth 2% of course grade.
  - A short list (a few bullet items) of what you liked and you disliked about the lecture.
  - You volunteer for a class, and must submit within 24 hours to get credits.

Re-grading

- Please meet with the TA or the instructor who was responsible for the work and arrange for regrading.
- **You have one week from the day grades are posted or mailed or announced.**
  - Late requests will not be entertained.

Academic integrity

- You can discuss general assignment concepts with other students: explaining how to use systems or tools and helping others with high-level design issues.
- You **MAY NOT share** assignments or other answers by copying, retyping, looking at, or supplying a file.
  - Assignments are subject to manual and automated similarity checking (We do check! and our tools for doing this are much better than cheaters think).
- If you cheat, you will be brought up on academic dishonesty charges - we follow the university policy:
  - http://www.stonybrook.edu/commcms/academic_integrity

Disability

- If you have a physical, psychological, medical or learning disability, contact the DSS office at Room 128 ECC. Phone 632-6748/TDD
- If you are planning to take an exam at DSS office, you need to tell me ahead of time for every exam.
- **All documentation of disability is confidential.**

Catastrophic events

- Major illness, death in family, …
- Formulate a plan (with your CEAS academic advisor) to get back on track
- Advice
  - Once you start running late, it's really hard to catch up

What do you need to get started?

- Go to Google Classroom

  https://classroom.google.com/u/2/w/NjAxOTcwNjMzMDda/t/all

  Or follow the link on course Web page.

  - One of today's homework: fill out the questionnaire

- Get the textbook.

Mathematically Speaking

Variables

- *Is there a number with the following property: doubling it and adding 3 gives the same result as squaring it?*
  - In this sentence you can introduce a variable to replace the potentially ambiguous word "it": *Is there a number x with the property that* $2x + 3 = x^2$*?*
  - A variable is a temporary name until we can find the possible value(s).
- *No matter what number might be chosen, if it is greater than 2, then its square is greater than 4.*
  - a variable is a temporary name to the (arbitrary) number you might choose enables you to maintain the generality of the statement: *No matter what number n might be chosen, if n is greater than 2, then* n^2 *is greater than 4.*

Some Important Kinds of Mathematical Statements:

- Universal conditional statement: *For all animals a, if a is a dog, then a is a mammal.*
- Universal existential statement: *Every real number has an additive inverse.*
- Existential universal statement: *There is a positive integer that is less than or equal to every positive integer.*

Sets

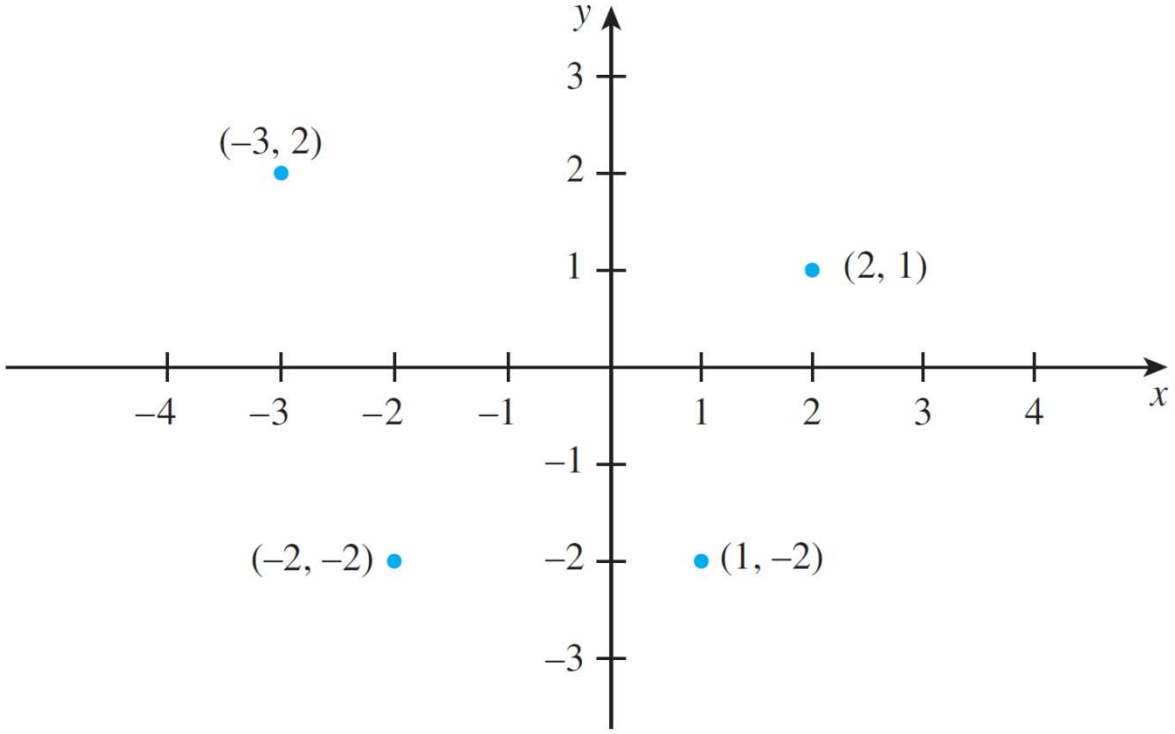
- Introduced in 1879 by Georg Cantor (1845–1918).
- A set is, intuitively, a collection of elements.
- Set-Roster Notation:
  - Let $A = \{1, 2, 3\}$, $B = \{3, 1, 2\}$, and $C = \{1, 1, 2, 3, 3, 3\}$.
    - What are the elements of A, B, and C?
    - How are A, B, and C related?
- Set-Builder Notation:

$$\{x \in \mathbf{R} \mid -2 < x < 5\}$$

- Subset: is a basic relation between sets : $\{2\} \subseteq \{1, 2, 3\}$

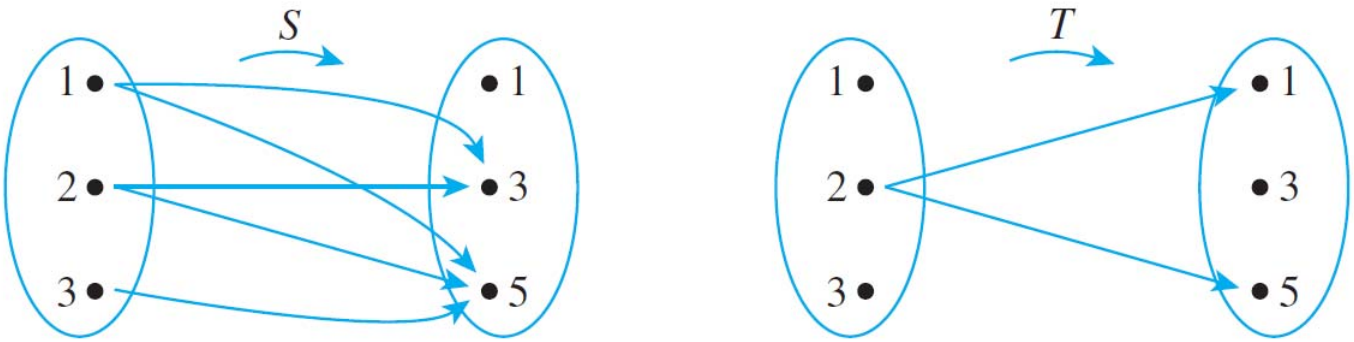
Cartesian product

- Example: **R** x **R** is the set of all ordered pairs (x,y) where both x and y are real numbers
- Cartesian plane:



Relations

- The notation $x\ R\ y$ as a shorthand for the sentence "x is related to y", for example: $1 < 2$
- From relations to sets: $x\ R\ y$ means that $(x, y) \in R$, for example: set $\{(2,1),(2,5)\}$
- Arrow diagrams of relations:



Functions

• Definition

A **function F from a set A to a set B** is a relation with domain A and co-domain B that satisfies the following two properties:

1. For every element x in A, there is an element y in B such that $(x, y) \in F$.
2. For all elements x in A and y and z in B,

$$\text{if} \quad (x, y) \in F \text{ and } (x, z) \in F, \quad \text{then} \quad y = z.$$

Example: The **successor function** g from **Z** to **Z** is defined by the formula $g(n) = n + 1$

A function as a machine: taking each input to a unique output.

Please

- Please be on time
- Please show respect for your classmates
- Please turn off (or use vibrate for) your cellphones

…

- On-topic questions are welcome

Welcome and Enjoy!