CSE215
Foundations of Computer Science
Course Information

Lecture 01 / Spring 2015
State University of New York, Korea
Instructor: Dr. Ilchul Yoon

Adapted from slides by Paul Fodor
Course Description

• “Introduction to the logical and mathematical foundations of computer science. Topics include functions, relations, and sets; recursion and functional programming; 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 in Stony Brook: AMS 151, MAT 125, or MAT 131.
General Information

- Meeting Information:
  - Lectures: Tu/Th 12:30PM - 1:50PM, B203
  - Recitation section: Friday 11:00AM - 11:53AM, A313

- During recitations, the TAs will reinforce lecture material and guide problem solving sessions.

- Course Web page

- Blackboard will be used for assignments, grades and course material
Instructor Information

- Dr. Ilchul Yoon
- Office: Academic Zone B #421
- Office hours: Monday & Wednesday 11:00AM – 12:00PM
  - I am also available by appointment

- Email: icyoon AT sunykorea.ac.kr

- Please include “CSE 215” in the email subject and your name in your email correspondence
TA Information

- Ayush Kumar
- Recitation hour: Tuesday 11:00AM – 11:53PM
  - TA will also be available by appointment

- Email: ayush.kumar AT sunykorea.ac.kr

- Please include “CSE 215” in the email subject and your name in your email correspondence
Course Description

- CSE215 is difficult: it is **very** demanding and is meant to be **very** demanding.

- Expect to spend about 20 hours per week outside of class on this course!!!

- There are challenging homework assignments and exams (2 midterms + final).
Getting Help

• Many of you will find CSE215 a stressful experience.

• Keep up with the reading assignments and homeworks!

• Don’t be afraid to see me and the TAs.

• Form a study group and work together to learn the material – not for copying homework solutions.
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 sequence of steps to perform a task must 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.
Textbook

- Discrete Mathematics with Applications
  Author: Susanna S. Epp
  Publisher: Brooks Cole; 4th edition (International)
Important Dates

- Midterm 1: Tuesday 3/31 @ B203 12:30PM-1:50PM
- Midterm 2: Thursday 4/30 @ B203 12:30PM-1:50PM
- Final Exam: Thursday June 19, 12:30PM-3:00PM

- The exams will be similar to what we solve in the class!
Coursework

• Grading Schema

<table>
<thead>
<tr>
<th>Type</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-Class Quiz (3~5)</td>
<td>100 pts</td>
</tr>
<tr>
<td>Homework (5~7)</td>
<td>140 pts</td>
</tr>
<tr>
<td>Programming Project (1)</td>
<td>60 pts</td>
</tr>
<tr>
<td>Mid-Term Exam #1</td>
<td>200 pts</td>
</tr>
<tr>
<td>Mid-Term Exam #2</td>
<td>200 pts</td>
</tr>
<tr>
<td>Final Exam</td>
<td>300 pts</td>
</tr>
<tr>
<td><strong>Maximum Points Possible</strong></td>
<td><strong>1000 pts</strong></td>
</tr>
</tbody>
</table>

• Letter Grades (Guaranteed)

A = [90-100]%, B = [75-90]%, C = [55-75]%, D = [30-55]%, F = [0-30]%
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:
  \[ \{x \in \mathbb{R} \mid -2 < x < 5\} \]
- Subset: is a basic relation between sets: $\{2\} \subseteq \{1, 2, 3\}$
Cartesian product

- Example: $\mathbb{R} \times \mathbb{R}$ 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 $(x, y) \in R$
- Arrow Diagrams of Relations:

\[ T = \{(2,1), (2,5)\} \]
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 \( B \),
   
   if \( (x, y) \in F \) and \( (x, z) \in F \), then \( y = z \).

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