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: AMS 151 or MAT 125 or MAT 131.
General Information

- Meeting Information:
  - Lecture section 2: Tuesdays and Thursdays 4:00PM - 5:20PM, Engineering 143.
  - Recitation section 8: Tuesdays 5:30PM - 6:23PM, Frey Hall 112.
  - Recitation section 9: Thursdays 5:30PM - 6:23PM, Melville N4000.
  - Recitation section 10: Tuesdays 7:00PM - 7:53PM, Chemistry 126.
  - Recitation section 11: Thursdays 7:00PM - 7:53PM, Physics P117.
  - Recitation section 12: Mondays 10:00AM - 10:53AM, Melville N3063.

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


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

- **Course Web page:** [http://www.cs.stonybrook.edu/~cse215](http://www.cs.stonybrook.edu/~cse215)
- **Blackboard** will be used for assignments, grades and course material.
- **Staff mailing list:** [cse215ta@cs.stonybrook.edu](mailto:cse215ta@cs.stonybrook.edu)
  - Use this for all communication with the teaching staff
  - Send email to individual instructors only to schedule appointments
Instructor Information

- Dr. Paul Fodor
  New Computer Science Building room 214
- Office hours: Tuesdays 11:30AM-1:00PM and Thursdays 2:30PM-4:00PM
  I am also available by appointment
- Email: paul (dot) fodor (at) stonybrook (dot) 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; 1st edition (2011)
  ISBN-10: 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 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.
Important Dates

- Midterm exam 1: Thursday, 3/03/2016, 4:00PM - 5:20PM, Engineering 143.
- Midterm exam 2: Thursday, 4/14/2016, 4:00PM - 5:20PM, Engineering 143.
- Final exam: Monday, May 16, 2016, 2:15PM - 4:15PM, in Engineering 143.
  - [http://www.stonybrook.edu/registrar/finals.shtml](http://www.stonybrook.edu/registrar/finals.shtml)
- The exams will be like what we solve in the class!
Coursework

• Grading Schema
  • Homework and class quizzes = 25%
    • Class quizzes
    • Homework assignments
  • Midterm exams (2) = 40% (20% each)
  • Final exam = 35%
Regrading of Homework/Exams

- Please meet with a TA or the instructor 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, source code 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/uaa/academicjudiciary](http://www.stonybrook.edu/uaa/academicjudiciary)
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?

- Blackboard account
  - [http://blackboard.stonybrook.edu](http://blackboard.stonybrook.edu)
- 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:
\[
\{ x \in \mathbb{R} \mid -2 < x < 5 \}
\]

- 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:
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 $B$,

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

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