Stuyvesant High School Computer Science
MKS21 - Introduction to Computer Science - One Semester

teacher:
email:
office: 301
available periods:

course website: stuyvesant-cs.gtihub.io/courses/intro0

Course Description:

Stuyvesant's two-term computer science introduces students to many major concepts in the field all while learning how to program. Students are given a strong foundation in computer science including an introduction to more advanced topics like recursion, state machines and cellular automata.

Required & Recommended Tools:

- Notebook/Section in binder.
  - All students are required to take physical (pen & paper) notes for this class (barring any required accommodations).
- Online Q & A forum.
  - Invitation links will be sent out after the first week of classes, you must accept.
- Recommended
  - Access to a computer outside of class.
    - Installation of DrRacket (https://download.racket-lang.org/) and NetLogo (https://ccl.northwestern.edu/netlogo/download.shtml) is needed.
  - The CS Dojo (room 307, M-Th, 3:45 - 5), provides access to a computer lab after school staffed with trained upper class peer mentors.
- When applicable, links to primary documentation for the tools used will be provided. Use of other online resources, such as stack overflow, should only be considered after checking primary documentation and the class Q&A forum.

Course Requirements:

- Treat each other with respect.
- Come to class on time.
  - Absences and latenesses must be accompanied by a note.
  - If you miss an exam, you must have a note before scheduling a make up.
- Participate in class discussions, including the online Q & A forum.
- Submit work on time.

Grade Breakdown:

- Participation: 10% (this includes in-class discussions, Q&A participation and group work).
- Work assignments: 20%
- Exams and Projects: 70%
  - There will be 4-5 regular in class exams and a departmental final exam.
  - There will be at least one project that will count as an exam.

Course Outline:

- **Unit I: Fundamentals of Programming using Racket**
  - Why do we have programming languages?
  - Syntax vs Semantics
  - Using Racket-style (prefix) notation.
  - Using functions in racket.
  - What is an algorithm?
  - Writing functions in racket. (lambda optional)

- **Unit II: Boolean Values and Conditional Statements**
  - What are Boolean values?
  - How do comparison operators work?
  - What are boolean operators?
  - Using functions that return boolean values.
  - Using `If`.
  - Using `Cond`.
  - Introduction to recursion.

- **Unit III: Computer Hardware/Software**
  - How do computers represent information?
  - What are the main parts of a computer, how do they work together?
  - What is the purpose of the operating system?
  - What is the internet? How are computer connected on the internet?
  - What can be done to secure your information and identity while online?

- **Unit IV: Introduction to NetLogo Programming with Turtles and Patches**
  - Understanding Agents and context (turtles, patches, observer).
  - Modifying turtle properties.
  - Modifying patch properties.
  - Representing colors.
  - Moving turtles.
  - Using random values.
  - Using boolean values: comparison operators, boolean operators, if and ifelse.
  - Writing procedures with and without parameters.

- **Unit V: Customizing the Interface and Advanced Procedures**
  - Using buttons and sliders in the interface.
  - Writing Reporter functions.
  - Using monitors.

- **Unit VI: Advanced Control of Agents and Age Sets**
  - Using with to filter agent sets.
  - Using ask to direct specific agents and agent sets.
  - Creating and using custom properties.
  - Creating and using breeds.
  - Turtle-Patch interaction

- **Unit VII: Working With Patches**
  - Patch properties.
  - What are cellular automata, including Conway’s Game of Life.
  - Programming Conway’s Game of Life and other life-like cellular automata.

- **Unit VIII: Building larger Programs**
  - Designing & Prototyping projects before programming.
  - Reviewing program prototypes.
  - Iterative program development with check points.
  - Demonstrating completed projects.

- **Optional Topics** (To be added at teachers discretion)
  - **Racket**
    - Using lists to store multiple elements.
    - Using map and filter to manipulate lists.
  - **NetLogo**
    - Using patches to manipulate images.
      - Edge detection
    - Turtle-Turtle interaction.
    - Using lists.
    - Navigating the models library
      - **Optional project:** Demonstrating a model from the models library.