

Computer Curriculum

Academic Advisory Committee

Why are we doing this?

Career in Programming –Fame and fortune

Media momentum-“Hour of Code” (beware of fashion!)

We used to have it in SMHS, let’s bring it back

Helps with Internships

Teaches problem solving skills

Demystify the world around us

Culmination of secondary school math education?

Scarsdale school district

(sister district?)

Profile:

- Affluent New York City suburb: about 350 students per grade
- “Historically a factory for lawyers and doctors, but now doctors and lawyers can’t afford to live there.”
- Strong track record of students admitted to selective colleges
- 50%+ Jewish, 5-10% Asian
- Extensive use of tutoring
- Budget funded by local property taxes--\$29k per student

Computer coding curriculum

- Two full year high school courses
- Nothing in elementary, middle school, but parents asking for it
- Also Interest in developing STEM tracks
- “Remember the Girls”
- Eg. David Siegel, chairman of Double Sigma, \$24 billion quantitative hedge fund
- <http://www.scarsdaleschools.org/site/default.aspx?PageType=3&ModuleInstanceID=14953&ViewID=7b97f7ed-8e5e-4120-848f-a8b4987d588f&RenderLoc=0&FlexDataID=20572&PageID=13467>

What Do Students Learn When They Code?

Coding Skill	Real Life Application
Sequencing	Ordering of tasks
Parallelism	Delegating work to a team
Debugging	Fixing problems efficiently
Modularization	Breaking a complex problem into smaller pieces

Source: Siegel talk, Scarsdale school district

Modularization



Empowerment



Issues

Computer systems vs coding

Which language?

Leverage online resources

- Flip classroom?

Prerequisites/Pace

Curriculum by grade level

Elective vs required

Computer Systems vs Coding

Computer Systems	Coding
<p><u>Under the hood:</u> To understand coding, helpful to understand the hierarchy of computer architecture:</p> <ul style="list-style-type: none">• Billions of tiny on/off switches• Boolean logic and binary arithmetic• Assembly language- primitive operations• Input/output• Networking <p><u>Big Picture technology literacy:</u> To understand the world around us, helpful to understand a variety of computer applications and technologies</p> <ul style="list-style-type: none">• Simplified introduction to algorithms	<p>Doing vs watching</p> <p>Mental skills</p> <ul style="list-style-type: none">• Sequencing• Parallelism• Debugging• Modules• Economy and clarity

Widespread use of computers is only about 50 years old
 We only have a generation or so of experience in teaching it.
 Pedagogy is in flux.

When	Computing environment	Programming Languages	Programming in general curriculum
1960-1985	mainframes, then minis Terminal to host	Assembly FORTRAN BASIC COBOL	none
1985 to 1995	PC's word processing, spreadsheets	C, 4 th generation	BASIC
1995 to 2010	PC's Internet	HTML Visual BASIC C++, JAVA (Object orientation)	PC Applications (Excel, Word,Powerpoint)
Last 5 years	Smartphones Internet	Python (and various others)	?

Types of programming languages

Object oriented languages invented to group code and data into reusable modules, so that it could be understood and maintained more easily.

4th generation code was intended to bring programming closer to the user.

	Procedural	Object oriented	4-GL
Elements	Variables Data types Operations Flow control Arrays Functions Recursion....	Everything in procedural, plus a rigorous abstraction of the whole system into modules called classes	Simplified tools and syntax Specialized purpose Typically shorter programs focused on analyzing data
Type of mental effort	Bottom up: Attention to the details of logic	Top down: Vision of how the whole may be divided into parts	Iterative analysis
Examples	BASIC COBOL FORTRAN C	Python JAVA C++	SQL (Databases) SPSS & SAS (statistics) MATLAB (Matrix operations)

AP question on Object Oriented simple, right?

6. A car dealership needs a program to store information about the cars for sale. For each car, they want to keep track of the following information: number of doors (2 or 4), whether the car has air conditioning, and its average number of miles per gallon. Which of the following is the best object-oriented program design?
- (A) Use one class, `Car`, with three instance variables:
`int numDoors`, `boolean hasAir`, and
`double milesPerGallon`.
 - (B) Use four unrelated classes: `Car`, `Doors`, `AirConditioning`, and `MilesPerGallon`.
 - (C) Use a class `Car` with three subclasses: `Doors`, `AirConditioning`, and `MilesPerGallon`.
 - (D) Use a class `Car`, with a subclass `Doors`, with a subclass `AirConditioning`, with a subclass `MilesPerGallon`.
 - (E) Use three classes: `Doors`, `AirConditioning`, and `MilesPerGallon`, each with a subclass `Car`.

Demo hour of code

<http://studio.code.org/hoc/1>

<http://studio.code.org/hoc/18>

Advanced K-5 course

<http://studio.code.org/s/course3/stage/13/puzzle/12>

Scarsdale High School

Introductory Computer Course

Computer 913

1. Introduction to Computer Science

- a. Introduction to Computer Hardware
- b. Evolution of Computer technology
- c. Evolution of Computer Languages
- d. Use of standards
- e. Memory Hierarchy
- f. Units of storage
- g. Basic Components of Computer System
- h. Processing Instructions
- i. Fetch-Execute Cycle (CU, ALU)

2. Using Excel-Intro to computer concepts

- a. Basic Excel Operations
- b. Concept of Data types
- c. Intro to Functions
- d. Return types
- e. Arguments/parameters
- f. Nesting Functions
- g. Looking at Some What If Excel features
- h. Filters
- i. Using Excel as a flat file database
- j. Pivot Tables
- k. Intro to Programming in VBA

3. Computer Number Systems and Information Encoding

- a. Understanding Positional Number Systems
- b. Binary, Hexadecimal, Octal
- c. Conversion between number systems
- d. Representation of negative numbers
- e. ASCII and other encoding

4. Digital Circuits/ Logic

- a. Gates
- b. Truth Tables
- c. Boolean Algebra and operations
- d. Theorems/Axioms of Boolean Algebra
- e. Simplification of Boolean Functions (Algebraic and Karnaugh Maps)
- f. Constructing Circuit Diagrams from Truth Tables
- g. Typical Circuits-i.e., Half Adders, Multiplexers, Decoders, 7-segment display

5. Intro to Data Modeling and SQL (using MYSQL)

- a. Modeling an Environment
- b. Identifying Entities /Relationships
- c. Entity Attributes
- d. Relationship Properties (Cardinality, Optionality)
- e. Constructing E-R diagrams
- f. Constructing Tables
- g. Constructing Queries
- h. Intro to SQL-SELECT, FROM, WHERE clauses
- i. Different types of joins (Cross products) and queries
 - i. i.e. aggregate functions
- j. Look at data related features (reports, mail merge, file formats)

6. Web Design/Development

- a. Introduction to the Internet
- b. Intro to DNS, Client Server Model
- c. Intro To Mark-up Languages
- d. HTML and evolution of XHTML
- e. Formatting the Webpage using CSS
- f. Programming (via PHP)
- g. Intro to Algorithm design
- h. Control Structures
- i. Creating Functions
- j. Typical algorithms
 - i. Sorting, Searching
- k. Creating Dynamic Webpages using HTML, PHP and MYSQL.
- l. 3-Tier architecture

Scarsdale High School

AP Computer -1

First unit is on classes and objects, then algorithms and syntax

Advanced Topics Computer Science 925 Course Outline

Modified Summer 2013

Textbook Java Methods A & AB by Litvin and Litvin (ISBN 978-0-9824775-7-1)

UNIT 1: Classes and Objects

Problem Solving via Microbot

- Classes vs instances/objects
- Introduction to GridWorld (part 1)
 - Object diagrams
- State of an object
- Designing your own class (on a pseudocode level)
- Structure of a class (fields, constructor(s), method(s))
- Signature of a method
- Method parameters
- Accessor and mutator methods

Textbook Sections

3.3
3.2
3.3

3.3, 3.5
3.5
3.6

UNIT 2: Algorithms and Introduction to Java

- Algorithms, pseudocode, and flow charts
- Summation notation (sigma notation)
- Java syntax and style
- Console output (`print` vs `println`)
- Literal values vs evaluated expressions
 - Variables and data types
- Assignment statements
- Scope of a variable (principle of narrowest scope)
 - String concatenation
- Boolean expressions, order of operators, De Morgan Laws
- Conditionals, short-circuit evaluation

Textbook Sections

4.1 – 4.2
supplement
5.1 – 5.7
2.4
6.4 – 6.5, 10.2
6.1 – 6.3
6.2
6.6
6.9
7.3 – 7.5
7.7 – 7.8

Scarsdale High School

AP Computer 2

UNIT 3: Loops and Iteration

- While loops
- Do while loops
- For loops
- Return and break in loops
- Nested loops

Textbook Sections

8.2
8.3
8.2
8.4
8.5

UNIT 4: Object Interaction

- Abstraction and Modularization
- Encapsulation and information hiding
- Public vs. private
- Object Diagrams
- Object Creation
- Using a user-defined class as a variable type
- Primitive types vs. Objects
- Method Calls
 - Internal method calls (should be declared as private)
 - External method calls (should be declared as public)
 - Dot notation

Textbook Sections

9.1

9.2

9.4
9.4

9.6

Scarsdale High School

AP Computer -3

-Immutability of Strings	10.3
-Javadoc comments	<i>supplement</i>
Use class documentation to alter a class	<i>supplement</i>
UNIT 6: Arrays and ArrayLists	<u>Textbook Sections</u>
-Arrays (1-D and 2-D)	12.2, 12.4, 12.7
-Write an array-based implementation of an ArrayList	
-ArrayList class	13.2, 13.3, 13.4
-For-each loops	12.6
UNIT 7: Searching and Sorting	<u>Textbook Sections</u>
Sequential search	14.3
-Binary search	14.3
-Selection sort	14.5
-Insertion sort	14.6
UNIT 8: Interfaces and Inheritance	<u>Textbook Sections</u>
Single inheritance	11.1, 11.2, 11.4, 11.5
-Polymorphism	11.7
-Interfaces	11.8
-Abstract classes	11.3
UNIT 9: Recursion	<u>Textbook Sections</u>
-Infinite, head, and tail recursion	4.1, 23.1
-Recursive searching algorithms (binary, sequential)	
Recursive sorting algorithms (mergesort, quicksort)	14.7, 14.8

PCC Curriculum Java

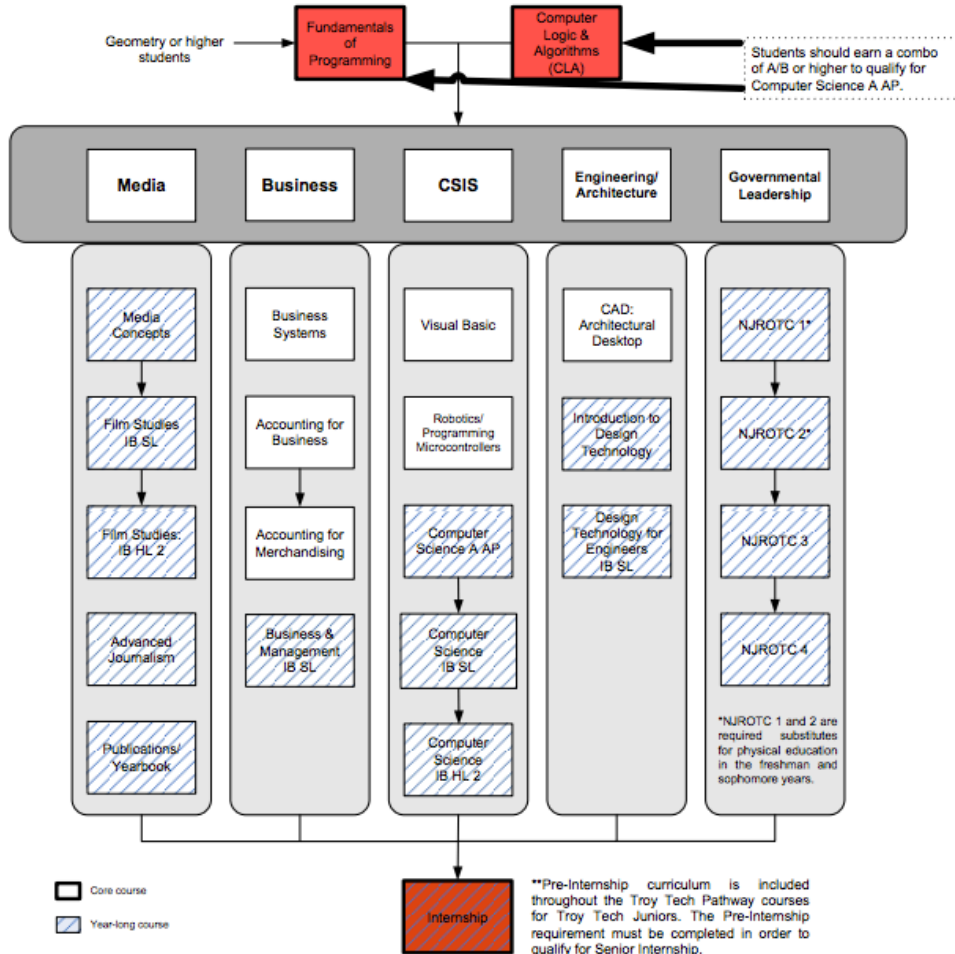
Prerequisites: 1-2 semesters of introductory courses
 Last quarter of course focused on game development

Weeks	Topics Covered	Homework/Assignments
Week 1 Aug 25	Syllabus, Eclipse basics	Introductions.Hello World program
Week 2 Sept 1	more Eclipse, graphics, Instance, Local vars	Draw Something
Week 3 Sept 8	Images, more variables, Instance, Methods, Swing, JLabel, If Else	Variables and Methods
Week 4 Sept 15	Classes, Images, JButton, JText, Check box, Radio buttons, ComboBox JList Component,	If Else statements
Week 5 Sept 22	GUI, formatting numbers, Layouts	GUI Design
Week 6 Sept 29	Variables and math, switch, Conv	Calculator Program
Week 7 Oct 6	Events, Loops	Switch Statement, JAR file
Week 8 Oct 13	Midterm,	Bubble Sort
Week 9 Oct 20	Classes, JTabbedPane, tooltips, audio, JFrame	Average and reverse an array
Week 10 Oct 27	Media, Arrays, JTable, ArrayList , Tabs	Hospital Patient Roster part 1
Week 11 Nov 3	Threads, Timer	Hospital Patient Roster part 2
Week 12 Nov 10 Tuesday Off	Games	Tower of Defense Game
Week 13 Nov 17	Sound and Polymorphism	Part 2 of Tower of Defense Game
Week 14 Nov 24 Thursday Off	Rectangles and collision detection, inheritance	Part 3 of Tower of Defense Game
Week 15 Dec 1	Preparing for final, review	Final Tower of Defense Game
Week 16 Dec 8	Final	

Troy High School

includes community high school and magnet school

Troy Tech Pathways 2014-2015



Students are not limited to one pathway, but may sample courses from several pathways.

SMHS FIRST Robotics

Students use four different professional packages of engineering software to design, build, and operate a robot

Engineer's drawing

- AutoCAD (Autodesk)

Robot Control

- LabVIEW- Graphical process control software(National Instrument)
- Java, using robotics library of classes and templates

Machine Vision

- Tools for recognizing shapes in pixel map(National Instrument)

Students teach each other, teach themselves.

But, only 8 weeks/year plus intermittently in fall

Online Resources

Codingbat: online practice for Java and Python using short exercises

Code.org: online materials for all levels K-12

Code Academy: short tutorials

UC Scout: AP computer science tutorial

Amplify: AP computer science

Others:

- Java website tutorial
- Khan Academy
- MIT, Harvard, Stanford etc computer science lectures on youtube

Code.org

K-5 (good, but bugs describe and review)

- Hour of Code
- 3 progressive online tutorials
 - 20 hours each
 - (sampled about half of them. some bugs, but responsive to bug reports)
 - Teaches algorithm development using blocks (but can peek at code generated)
- Teacher dashboard to monitor student progress, teacher workshops

Middle School Modules (not reviewed)

- Computer Science Math
- Computer Science Science

High School – 2 full year courses

- Exploring Computer Science , (cursory review)
 - “Democratize computer science”
 - Scratch, Web design using HTML, Javascript
 - Too elementary
- AP Computer Science Principles
 - Under development, appears too elementary

AP Computer Science

UC Scout

UC developed

2 Levels

- Free, no support, intended as enrichment or supplement
- \$149 per student per semester, UC provides teacher

31 lessons, beginning with objects and classes

Code Academy

Short tutorials, all but Python focused on web development
Includes tutorials for teachers and student tracking help

Free:

course	time	# of subscribers
HTML/CSS	7	4.5
Javascript	10	5M
JQuery	3.5	.5
PHp	4	.9
Python	13	3.5
Ruby	9	.8

Choice of language for an introductory course

	Python	Java	C++
+	<ul style="list-style-type: none"> Simplest syntax makes it easiest Introduction to object orientation along with algorithms Displacing Java as introductory program (MIT, Berkeley) 	<ul style="list-style-type: none"> Language of AP course Accommodates beginner errors well Rich library to explore many areas of CS (networking, threading, robotics) Most popular (e.g .Android smartphone) 	<ul style="list-style-type: none"> Most powerful, most efficient language, “Close to machine” “You are not a fully prepared computer professional until you are well versed in a language that is close to the machine, Introductory course can defer learning about object orientation.
-	<p>AP requires new language, (closer than C++ though)</p>	<ul style="list-style-type: none"> Object orientation required from start, which “to a beginner looks like a solution looking for a problem”, Java hides details of machine, so “when things go really, really wrong, ..Java programmers not as skilled in debugging ” 	<ul style="list-style-type: none"> Transition to AP not so easy Difficult concepts <ul style="list-style-type: none"> Memory management Arrays as pointers “Starting out with C++ is like learning to cook in an industrial kitchen”

Source: Quora and Stackoverflow forums

<http://www.quora.com/Which-is-more-beneficial-to-learn-as-a-first-programming-language-Java-or-C++-Why>

<http://stackoverflow.com/questions/1085134/why-is-c-relatively-harder-to-use-bad-choice-for-a-beginner>

Prerequisites?

AP level courses in Scarsdale, Troy, PCC, and SMHS (Armbruster)

- 1 or 2 semesters of introductory coursework,
- including an overview of computer systems and/or an introduction to coding.

College Board AP course mentions only these prerequisites

Students should have knowledge of basic English and algebra. Students should be comfortable with functions and the concepts found in the uses of function notation, such as $f(x) = x + 2$ and $f(x) = g(h(x))$. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course.

Preliminary Recommendations

artists conception

Elementary School-curriculum

- 10 to 20 hours/year of coding grades 3 to 5
- using online tutorials at Code.org
- possibly concentrated in May, after testing

Elementary –after school/summer enrichment

- SMCSO offers app developments thru Freshi Medi??
- Enhance this, eg Mindstorm

Middle School

- Elective: 7th or 8th grade: Python
- Code academy course first 3 weeks

High school- electives

- Introduction to Coding (1 semester and summer school) Java
- AP based on UC Scout (year
 - Introductory course recommended, but can be waived
- Ideally-- crash course in MATLAB -zero or 6th period, evenings