USA: the Advanced Placement curriculum ”Computer Science Principles” (AP-CSP)

Andrea Sterbini – sterbini@di.uniroma1.it

USA: AP Computer Science Principles

AP: Advanced courses for High School students (==> credit 4 uni.)

Computational Thinking **practices** **vs. main topics**

P1: Connecting Computing

P2: Creating Computational Artifacts

P3: Abstracting

P4: Analyzing Problems and Artifacts

P5: Communicating

P6: Collaborating

Big Idea 1: Creativity

Big Idea 2: Abstraction

Big Idea 3: Data and Information

Big Idea 4: Algorithms

Big Idea 5: Programming

Big Idea 6: The Internet

Big Idea 7: Global Impact

USA: Many versions of the AP-CSP curricula available

Curriculum	Course Delivery	Programming Language / Environment
CodeCombat	Web Based	**JavaScript / Python / HTML**
The Beauty and Joy of Computing	Web Based edX	**Snap!**
Mobile CSP	Web Based	**App Inventor**
UTeach CSP	Web Based	**Scratch / Processing**
PLTW CSP	Canvas LMS Printable Student Content	**Scratch / App Inventor / Python / HTML**
Code.org CSP	Web Based	**App Lab / JavaScript (Blockly)**
CS50 AP	Wikispaces	**Scratch / C**
CS Matters	Face to Face	**Python**
EarSketch	Web Based: make music	**Python / JavaScript**
CodeHS	Web Based	**JavaScript**

The BJC curriculum (Beauty and Joy of Computing)

Unit 1: Introduction to Programming

Unit 2: Abstraction

Unit 3: Data Structures

Practice CREATE TASK

Unit 4: How the Internet Works

Unit 5: Algorithms and Simulations

CREATE TASK **<== EXAM**

Unit 6: How Computers Work

Unit 7: Fractals and Recursion

Unit 8: Recursive Functions

Unit 1: Introduction to Programming

ORGANIZATION: 5 Lab units (plus some optional)

Pair programming: Students work in pairs and swap role during the unit

Discussion of what to do as a way to enforce **ANALYSIS** before implementation

1) move a sprite randomly, greet, save the program

2) Gossiping Sprites: use functions to select a random message to "say", define functions, ask something

3) Polygons: draw, repeat, ask numbers

4) Protect Privacy ***(focus on social issues)***

5) Follow the mouse or another sprite

Optional projects: Pong, drawing, random sentences,

Unit 2: Abstraction

1) Variables: local (number guessing game) and global (score of the game), Import/Export blocks

2) Lists: shopping list app, quiz app

3) Making decisions: If-the-else, Predicates, Boolean expressions, list filters

4) Math library: making new math functions

5) Copyright and Fair Use ***(focus on social issues)***

Optional: modelling language (plurals), mastermind, kaleidoscope, automated fortune teller

NOTICE: the suggested programming style is FUNCTIONAL

Unit 3: Data Structures

1) Complex drawings (cycles)

2) ADT: managing a contact list (name surname phone number …),
by defining its builder and getters/setters

3) Tic-tac-toe: check for winning game, lists comparison, map

4) Robots and AI: introduction and implications to Society

5) Computers and work: new type of jobs, impact on work

Optional projects: drawings, animations, music

Unit 4: How Computers Work

1) Computer Abstraction Hierarchy:

Network redundancy, internet addresses, history

2) Cybersecurity, cryptography:

the Caesar cypher project

3) Social networks, cyberbullying, censorship, search engines

4) Data representation and compression

Unit 5: Algorithms and Simulations

1) Search algorithms and efficiency

2) Models and simulations: distributions of flipping a coin, spread of a virus, bank queue

3) Analysing data:

4) Unsolvable and Undecidable problems, Paradoxes, the Halting problem

5) Computer and Wars: cyberwar, drones, autonomous weapons, ethics

6) Tic-Tac-Toe with a Computer Player

EXAM (CREATE TASK)

AP CREATE TASK (exam practice)

Kids practice how to organize the design and development of the final “AP create task exam” with the help of teachers and peers

1) Using a Development Process to Organize Your Coding

2) Choosing Your Project

3) Implementing Your Development Process

4) Testing Your Project

5) Communicating About Your Project

6) Evaluating Your Work

During the exam they will have to work by themselves

Unit 6: How Computers Work (optional)

1) Computer abstraction hierarchy (10 levels)

Application
Programming Languages
Libraries
Operative System
Hardware
Components
Integrated Circuits
Gates
Transistors

2) History and Impact of Computers

Unit 7: Fractals and Recursion (optional)

1) Trees in a Forest

Recursive case

Base case

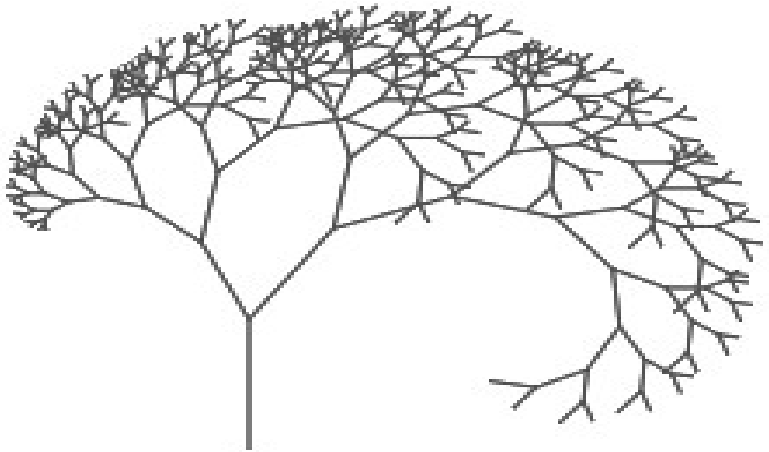
2) Recursion Projects

Sierpinski Fractal Triangle

Koch Snowflake

Lévy C-Curve Fractal

Recursive Mondrian



Unit 8: Recursive Functions *(optional)*

1) Recursive Reporters (functions)

2) Base conversion

3) Subsets

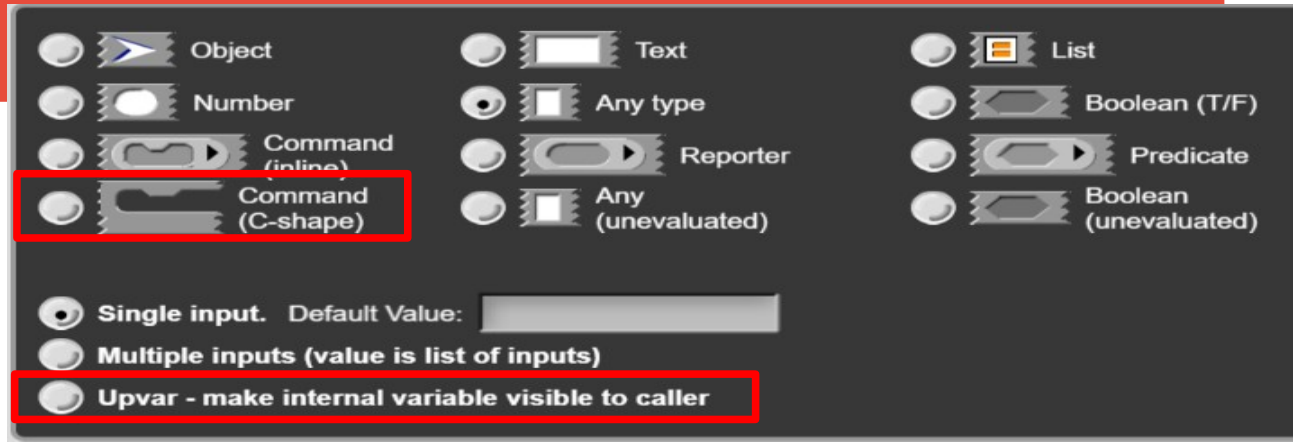
4) Higher Order Functions (on lists)

Optional Projects: Pascal/Tartaglia triangle, Sorting

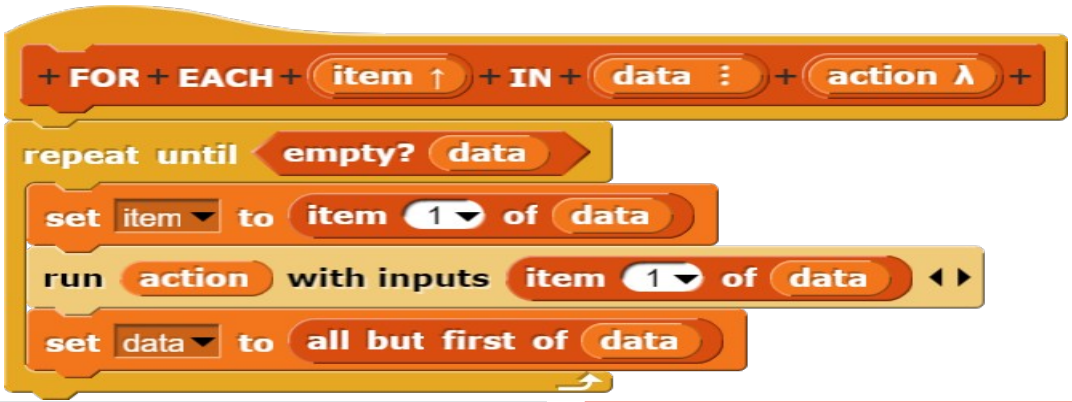
Snap!
Other features

Code arguments!

Arguments by reference!



This allows building meta-programming blocks/functions!



Robot maze exploration example