Intro to Computational Thinking for STEM -- Core

Course Description

This course introduces students to the basic ideas of computational thinking and its applications to problem solving in STEM fields. Students will use an open source, Web-based programming environment to create code for simple drawings, animations and simulations, through which they learn how to use abstraction, decomposition and pattern recognition to model problems and arrive to an algorithmic solution. Program code is presented with a dual purpose: as the main way to interact with a computer and as a proxy to organize ideas explicitly and communicate them to other people. Students taking Algebra I concurrently with this course will benefit the most, because many examples are drawn from Algebra I, so that students can visualize and manipulate the mathematical concepts in a more concrete form.

Course Objectives

- Describe applications of Computational Thinking to solve Math, Science and Engineering problems.
- Model objects made of multiple parts, as well as their behaviors and interconnections, using variables and functions, and construct virtual artifacts that simulate them.
- Use the program development process to create, debug, and redesign computing artifacts.
- Implement creative projects in which Computational Thinking and code is used to create artistic or technical renderings of diagrams, illustrations or graphs.
- Demonstrate the use of code as a medium to communicate ideas and designs precisely.
- Demonstrate effective communication skills, through team working, oral presentations, and good written communication.

Assessing Performance

Formative assessment includes worksheets and several practice activities for each lesson, and unit quizzes. Summative assessment includes a programming project at the end of each unit.

Course Essentials

Equipment	Cost/Unit
Classroom set of computers	$0 if you already have some, $500-600 per computer if you need to purchase

First Semester

Unit 1: Computing and Coding Basics	Computers and networks. Ethics of online communication. Basic Web design. Coding environments. Syntax and semantics of programming languages. Debugging techniques. Draw rectangles, squares, circles and polygons.
Unit 2: Transformations and composition	Overlays and Translations. The algebra of graphical transformations. Combine parts to create complex objects.
Unit 3: Symmetry and regularity	Scalings and Rotations. Compose operations to create regular patterns (stars, regular polygons). Design a clock using conjugation of rotations and translations. Exploit symmetry to create mosaics and quilts.
Unit 4: Managing complexity	Problem decomposition. Hierarchical organization of code. Heuristics. Diagrams.

Second Semester

Unit 5: Data and calculations	Lists and tuples. Random numbers. Text processing. Calculations with integers and dollar amounts. Charts.
Unit 6: Modeling with functions	Functions as models. Encapsulation and generalization. Degrees of Freedom. Dependent and independent variables. Constraints. Function transformations.
Unit 7: Patterns and repetition	Selection and piecewise functions. Recursion and iteration. Functions and graphs. Linear and rotational motion. Fractals.
Unit 8: Virtual artifacts	Animations and simulations. Periodicity. Sequential and parallel composition of animations. Internal state. If time permits: Create simple games. Collision detection.