TEALS Intro CS Curriculum Guide

As of 8/12/2014

TEALS Intro to CS is based on The Beauty and Joy of computing from UC Berkeley by Prof. Dan Garcia and his team. You can find their curriculum map, with links to source materials at: https://inst.eecs.berkeley.edu/~cs10/fa13/. Their class moves more quickly than the high school version, and sometimes takes topics in different orders. You can supplement those materials with materials from the Curriculum Materials library on this site, or other materials you find or design yourself.

Note on unit length: "your mileage will vary." It's more important for the students to master the content than for you to stick to the unit lengths below.
It's ok to cut one or more units if your students need more time.

Title	Length	Summative Assessment
Unit 1: What is computing?	2 weeks	Pen and Paper Test

Students are introduced to the study computer science, the SNAP/BYOB/Scratch programming environment, and basic flow-of-control commands.

Programming Objectives	Conceptual Objectives
• Identify by name the parts of the SNAP/BYOB IDE • Give examples of blocks in each category • Explain the function of common blocks (if, if/else, repeat, forever, pen down/up, …) • Determine the function of unfamiliar blocks • Add, remove and reposition sprites • Open/Save their work as specified (cloud or local storage)	• Define computing • List the functional parts of a computer and explain what they do • Determine whether various everyday objects contain computers • Create orderly instructions for solving a problem

Recommended Activities

- Fill-in-the-blank parts of the IDE
- SNAP Sleuth: explore unfamiliar blocks in SNAP and write hypotheses about what they do
- Greeting Card / Movie scene

Title	Length	Summative Assessment
Unit 2: Interactivity	2 weeks	Mario Project

Students continue to develop knowledge of procedural programming, with a focus on input/output. Students begin using programming as a way to model the world.

Programming Objectives	Conceptual Objectives
• Create or compose appropriate costumes for sprites • Animate a sprite in response to keyboard input • Use if/then/else blocks to make decisions in code. • Design and implement a basic model of gravity • Use loops to create repeated control structure	• Give examples of software that models aspects of the real world • Develop an abstraction

Recommended Activities

- Falling turtles lab
- Guess a number
- Choose your own adventure

Title	Length	Summative Assessment
Unit 3: Variables and Functions	3 weeks	Pong

Driven by the goal of a fully-functional interactive game, students learn to use variables, loops and functions, and begin to think about how to ensure performant behavior in programs.

Programming Objectives	Conceptual Objectives
• Create, store data in, and consume data from variables • Animate an object based on conditions of its state and environment • Analyze the performance of their program and find ways to optimize it • Create and call custom functions with descriptive names	• Compare and contrast the usage of variables in programming with variables in math • List a few common data types • Evaluate Boolean expressions • Describe hardware and software factors that influence performance (speed) in computing

Recommended Activities

- Rock, Paper, Scissors
- Brick Wall

Title	Length	Summative Assessment
Unit 4: Algorithms and Data	3 weeks	Hangman

Students learn to think algorithmically and to classify algorithms by their type and running time. Students start using lists to store and retrieve data.

Programming Objectives	Conceptual Objectives
• Create, populate, and consume data from lists • Design and implement an algorithm for searching for data in a list • Create and call custom functions that use arguments to influence output	• Explain what an algorithm is and give examples of various types of algorithms • Analyze the efficiency of an algorithm

Recommended Activities

- CSUnplugged: search and sort, puzzles, mazes

Title	Length	Summative Assessment
Unit 5: Objects	4 weeks	Space Invaders

Students learn about the Object-Oriented paradigm and use cloning in BYOB/SNAP to hierarchical object models based on a template.

Programming Objectives	Conceptual Objectives
• Dynamically create and destroy objects • Create objects that inherit behaviors from a parent • Compose a program that uses many different sprites with distinct behaviors to create an immersive experience	• Explain the difference between procedural and object-oriented programming • Give examples of situations in which OOP is appropriate

Recommended Activities

- Method-based communication
- Petri Dish

Title	Length	Summative Assessment
Unit 6: Development Practices	4 weeks	Self-Designed Project

Students channel the skills and knowledge they've developed to design, scope and implement their own project.

Programming Objectives	Conceptual Objectives
• Write a functional specification for a program • Decompose a large project into multiple checkpoints • Write pseudocode to outline the solution to a programming problem	• Use programming as a creative outlet • Evaluate a program for correctness.

Recommended Activities

- Reverse-engineered spec

Title	Summative Assessment
Unit A: Computing and Society *(horizontal)*	Current Events Presentation

Throughout the span of the class, students will be exposed to a variety of current, relevant stories that show the breadth of the computer science field. Students will consider legal, ethical and societal effects of digital technology in today's world.

Programming Objectives	Conceptual Objectives
• Tell a story with data	• Connect computing innovations with other fields • Analyze the beneficial and harmful effects of computing • Explain how technology impacts their life • Give examples of fields and careers that use computer science

Recommended Activities

- Journal Entries
- (Active watching) Video
- Blown to Bits
- Student presentation