**Daily Schedule**

**Monday**

**Morning class: Introduction** (180 minutes)

Objective: Coders will get comfortable with each other, the instructor, and the format of the course. Coders will understand the basics of Snap.

* (~15 minutes) Encourage the coders to get to know each other with a few ice breakers
* (~5 minutes) Discuss why they are here and what they would like to know by the end of the week
* (~5 minutes) Give the students an idea of the schedule for the week
* (~10 minutes) Classroom expectations: Let the students know what you expect from them - raising hands, being respectful to one another, collaborative not competitive, etc. Expectations on the computer: staying on the site the teacher tells you to be on. Encourage students to ask questions when they hear words they don’t know or get lost during a presentation.
* (~5 minutes) Start with a discussion of computers and computer science. Ask students about all the different things they can do on a computer. How does the computer know how to do those things? Prompt for: someone programmed it, someone coded it. Did they program all those things in the same language? No. Different languages are used for different things. Ask for any programming languages students are familiar with.
* (~30 minutes) Offline activity: Graph Paper Programming. Description and files to print are in the Monday folder, in the “Graph Paper Programming” folder.
* (~20 minutes) Introduce Snap (<http://snap.berkeley.edu/snapsource/snap.html>). Have students sign up for accounts, give them some time to play on the site.
* (~90 minutes) Work through the “Snap: Hour of Code”(<http://snap.berkeley.edu/hoc/#1>, must be on Google Chrome) with the coders. Start all together and then ask them to continue on their own. Let them know that this is jumping in and that there will be many things in this activity that they don’t know yet. They should make their best guesses, help each other, and try different things when it isn’t working. Instructor should pull up the solutions for each level to help struggling coders.
* If coders finish the Snap Hour of Code before the class period is done, have them work on this: <https://lightbot.com/flash.html>

**Afternoon class: Snap Map and Loops** (180 minutes)

Objective: Coders will get to know the Snap interface. Coders will understand what loops are and how they are used in Snap. Students will gain confidence in Snap and take more risks and get comfortable with the coding process.

* (~15 minutes) Give the coders a tour of the different areas of Snap (A map is provided in the Monday folder, “Snap Map.png” file)
  + Palette: Contains all the blocks we need for coding. Drag and drop them to the Scripting Area to make scripts. 8 different categories which we will go over throughout the week.
  + Scripting Area: Where you make your code by Snap-ing the blocks from the palette together
  + Stage: Where it shows your code running
  + Tool Bar: Where you can log into your account and save projects - show how to do this!
  + Sprite Corral: Where you keep track of sprites (you can have more than 1!)
* (~20 minutes) Offline activity: Loops. Description in the “Monday folder, in the “Loops” folder, “Loops Activity” file.
* (~10 minutes) Show coders the loop blocks available in the palette (in the Control section, Snap blocks called forever, repeat [], repeat until <>) and demonstrate a few ways to use each one
* (~15 minutes) Offline activity: Conditionals with Cards. Description and materials in the Monday folder, in the “Conditionals” folder.
* (~10 minutes) Show coders the conditional blocks available in the palette (in the Control section, Snap blocks called if<>, if<> else<>) and demonstrate a few ways to use each one
* (~15 minutes) Go over all the blocks in Pen, Sounds, Looks, and Operators sections of the palette
* (~50 minutes) Have the coders do Assignment 1: Loops and Conditionals in Snap
  + Solution to Loops: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%201%3A%20Conditionals>
  + Solution to Conditionals: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%201%3A%20Loops>
* (~45 minutes) Have the coders do Assignment 2: Introduce Yourself! In Snap
  + Solution: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%201%3A%20Loops>

**Tuesday**

**Morning class: Math in Snap** (180 minutes)

Objective: Coders will be able to graph, understand variables, and understand degrees, as well as use those concepts in snap.

* (~30 minutes each, ~90 minutes total) There are 3 math topics that need to be covered for Snap. Worksheets for each topic are in the Tuesday folder.
  + Graphing in the cartesian plane (Image in the Tuesday folder). Teach the coders how to graph an (x,y) point, identify the x-axis and the y-axis, and figure out the coordinates from a given point. Have the coders do Assignment 3: Graphing Puzzle in the Tuesday folder.
  + Show the Snap Stage Graph (Tuesday folder, file “GraphOfSnapStage.png”) and then move to Snap and graph-related blocks in the Motion section of the palette ( Snap blocks called: go to x: (), y: (), go to [], glide () secs to x: () y: (), change x by (), set x to (), change y by (), set y to ()), and demonstrate a few ways to use each one.
  + Angles: Introduce the 360-degree circle, right angles, 180-degree straight lines, and acute and obtuse angles. Show how to measure angles using a protractor. Have the coders do Assignment 4: Angle Types and Measuring in the Tuesday folder.
  + Show the angle-related blocks on Snap (in the Motion section of the palette, Snap blocks called: turn left () degrees, turn right () degrees, point in direction ()), and demonstrate a few ways to use each one
  + Variables & Functions: introduce this pre-algebra concept with care, as many 4-6th graders haven’t seen it before. Variables are symbols (usually letters in math, often words in programming) that stand for numbers in math, or many different possible types of information in programming. Functions take in a variable (or more than one), do some computation or other process, and return a result. Have the coders do Assignment 5: Functions and Variables in the Tuesday folder.
  + Show the Variables section of the palette, demonstrate ways to use each block
* (~90 minutes) Have the coders do Assignment 6: Shapes and Kaleidoscope in Snap. When they start the kaleidoscope part, show them the stage of the solution so that they can see what they are trying to create.
  + Solution to Shapes: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%206%3A%20Shapes>
  + Solution to Kaleidoscope: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%206%3A%20Kaleidescope>

**Afternoon class: Web Design 1** (180 minutes)

Objective: Coders will get to know the Google sites interface and the different aspects they can achieve with it. Coders will design their first website.

* (~5 minutes) Introduction: This is a change of gears from the program, so make sure to clarify that we’re changing topic for the afternoon. We will use Google sites, which is free and easy to use from any browser.
* (~45 minutes) Go over the demo site with the coders: <https://sites.google.com/site/educationunlimitedwebdesign/>. Then, copy the site and edit everything about it. Make sure to include changing things from the “Manage Site” section under settings, where you can change the background image and colors, font, font size, and font color, header format, page format and settings. You can edit the page orders and hierarchy here as well. Then, go back and edit a page, add a page, show how the navigation works. Show them how to edit text fields, images, and page names, how to create a new page.
* (~40 minutes) Snap Site: Pair the coders up to create a site about Snap. Ask them to include pictures, instructions, and example code!
* (~60 minutes) Duplicate Site: have the coders try to make a site that looks like one of your favorite websites (Facebook, etc)
* (~30 minutes) With the remaining time, allow to coders to create a site of their choosing. Let them know that this will become their final project for Web Design, and that they will have another 3 hours to work on it later in the week. Possible topics: a team page for the class, a demo of knowledge they learned in an elective or another class, an artwork page, another duplicate site, or whatever they want! Final project must include:
  + Images other than the theme images
  + At least 3 pages
  + A table of contents/navigator
  + A calendar, form, survey, or another gadget
  + At least 2 separate paragraphs of writing

**Wednesday**

**Morning class: Functions, Costumes, and Events** (180 minutes)

Objective: Coders will understand the role of events and be able to use them in Snap. Coders will work with Sprite costumes and costume changes, as well as sound effects, in Snap.

* (~30 minutes) Do offline Events activity in Thursday folder, show events on Snap Control section (they’ve already been working with them, but now have a name for it).
* (~30 minutes) Go over costumes and costume changes for Sprites in Snap, and broadcasting and receiving messages.
* (~60 minutes) Have the coders do Assignment 7: Costumes.
  + Solution: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%207%3A%20Costumes>
* (~20 minutes) Go over everything in the “Sound” section of Snap.
* (~40 minutes) Have the coders do Assignment 8: Sounds.
  + Solution: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Assignment%208%3A%20Sounds>

**Afternoon class: Animated Story** (180 minutes)

Objective: Coders will combine all that they know so far in Snap to “animate” a story using Snap. They will demonstrate their knowledge of what they’ve seen so far and build on it.

* (~15 minutes) Show example project “Wicked” in Snap: an aminated, shortened retelling of the musical Wicked. Then, walk them through the code for it (in 3 sprites and the background).
  + Wicked: <http://snap.berkeley.edu/snapsource/snap.html#present:Username=alixfeinsod&ProjectName=Wicked>
* (~165 minutes) Coders have today to work on the Assignment 9: Animated Story. If they finish early, ask them to add more detail or make a new one for another story. Let them know that if they don’t finish today, they can have more time to work on it tomorrow. Remind them to save the files frequently!

**Thursday**

**Morning class: Web Design 2** (180 minutes)

Objective: Coders will use all they’ve learned about web design to create a Google Site of their own design.

* (~180 minutes) With the remaining time, allow to coders to either continue with the site they started a few days ago, or create a different site, for their final Web Design project. Requirements: site must not have images and pages directly from a template. Site must have:
  + Images other than the theme images
  + At least 3 pages
  + A table of contents/navigator
  + A calendar, form, survey, or another gadget
  + At least 2 separate paragraphs of writing

**Afternoon class: Final Snap Project** (180 minutes)

Objective: Coders will use all they’ve learned about web design to create a final Snap program.

* (~180 minutes) Coders will work on their final projects. They can choose to continue the Animated Story as long as they make it detailed. A list of other choices for final projects is in the Thursday folder. Student-chosen projects are allowed provided they fulfil the following guidelines:
  + Are “showable” and interactive
  + Use at least 3 of the following: x-y coordinates, functions, degrees, conditionals, events (“when”), pen down-pen up, costumes, broadcasting messages
  + Take the whole time to create

**Friday**

**Morning class: Final Presentation Prep** (180 minutes)

Objective: Coders will choose to present their website or Snap project and create and practice their presentation.

* (~5 minutes) Today, we will be making and practicing our final presentations for families which will happen later today. If you present your website, you must do a walkthrough of all the pages and show use of all the features. If you present your Snap project, you must walk through all the scripts and how they work in your project. If the project is interactive, either demonstrate it or call a volunteer to use/play it.
* (~60 minutes) Coders work on their presentations
* (~45 minutes) Coders break into small teams and practice presenting each other and giving constructive feedback
* (~70 minutes) Coders can continue to work on their presentations, put finishing touches and additional features on their projects, or work on a challenge assignment for Snap (in Friday folder).

**Electives**

Elective 1: How Computers Work activity in Electives folder

Elective 2: Graph Paper Maze Solving in Electives folder

Elective 3: Requires computers! Code.org is an introduction to programming for non-coders. Students can do any hour of code tutorial from this list: <https://code.org/learn>

**Sources**

1. <https://teals-introcs.gitbooks.io/introduction-to-computer-science-principles>
2. <https://code.org/>