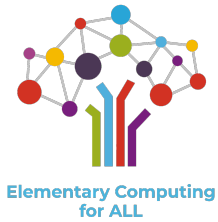


Introduction to the Elementary Computing For All Curriculum

The ECforALL Curriculum is an introductory Scratch curriculum that is designed to teach computational thinking to English language learners and culturally and linguistically diverse students in 4th-6th grades. It draws on the Creative Computing curriculum developed by the ScratchEd team at the Harvard Graduate School of Education including elements from Code.org, Wonder Workshop, and Bootup.org. The curriculum was then further developed by the ECforALL team -- including San Francisco Unified School District's CSinSF.org; the Digital Learning Lab at the University of California, Irvine; the CANON Research Lab at the University of Chicago; Texas State University; and teachers and administrators in Santa Ana Unified School District -- with funding from the National Science Foundation and the US Department of Education. It can be accessed at bit.ly/ecforall.

The curriculum is designed to provide support for teachers new to Scratch and computer science (CS) concepts, but is openly available and encouraged to be used and adapted by any teacher who is eager to introduce CS concepts to their students. The CS concepts covered that are explicitly taught are algorithms, sequences, events, and loops. Many introductory block-based computational thinking curricula exist; what differentiates this curriculum is the intentional integration of strategies suggested by the National Academies of Sciences that support English language learners in their competence in computer science. This curriculum aims to entice learners of English language learners into the realm of computer science to provide them with access to the vocabulary, concepts, practices, and skills that can enable them to become a part of the CS community.

Pedagogical Influences

In 2018, the National Academies of the Sciences published a report that examined the research on K-12 ELs' learning, teaching, and assessment in STEM subjects. Among the findings discussed in this report were the following 5 promising instructional strategies to support STEM content and language development.

- Engage students in CS practices
- Engage students in meaningful interaction

- Use multiple registers and modalities
- Leverage student resources
- Focus on how language functions in CS

For details of what these strategies look like in the classroom and how they have been addressed in this curriculum, click on this link.

In addition to incorporating these ELL STEM strategies to scaffold the CS instruction, effective pedagogical models in STEM are embodied in this curriculum. These models include Use-Modify-Create, 5e instructional model, TIPP&SEE, and Universal Design for Learning.

Use->Modify->Create

The Use->Modify->Create pedagogical approach is followed in each unit to gradually release independence to students. First, a CS Concept is introduced using an Engage activity that connects to the students' lives and an Unplugged activity to motivate students. Students then **Use** a pre-created Scratch project to observe the concept from a “user” lens and explore the code for the corresponding actions from a “coder” lens. Next, students **Modify** (“Quest”) the same project to practice coding and increase their understanding of the CS concept. Finally, students **Create** an open-ended project that includes requirements for coding the focal CS Concept.

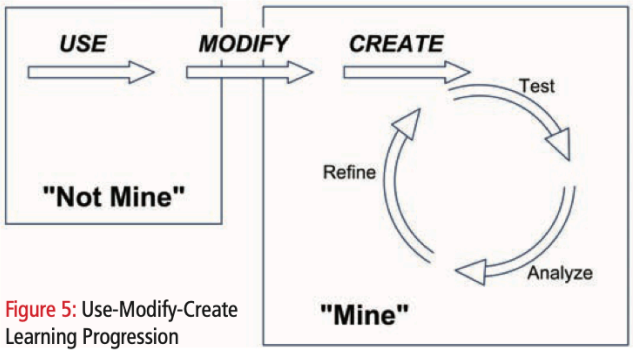
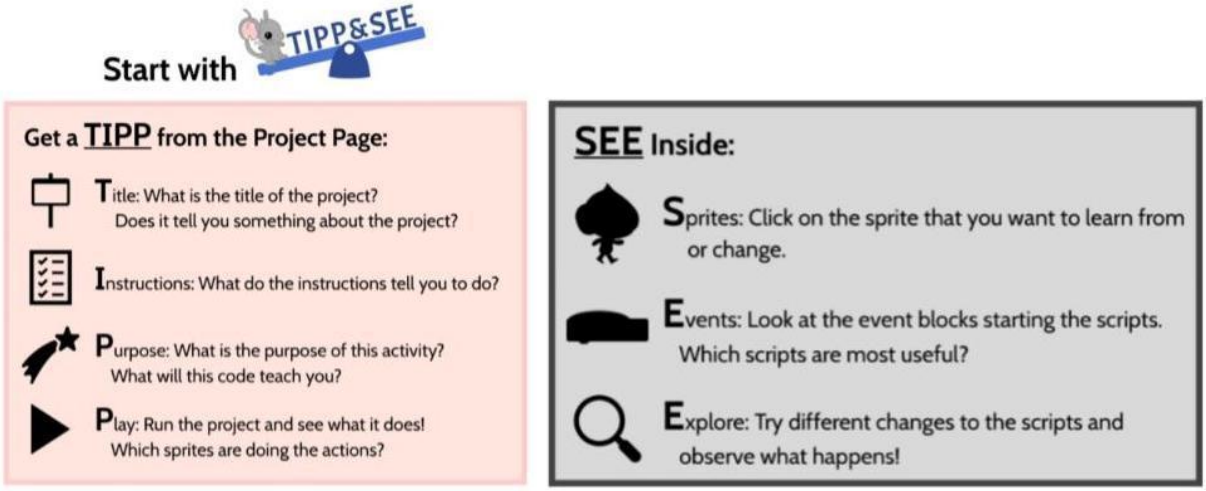


Figure 5: Use-Modify-Create Learning Progression

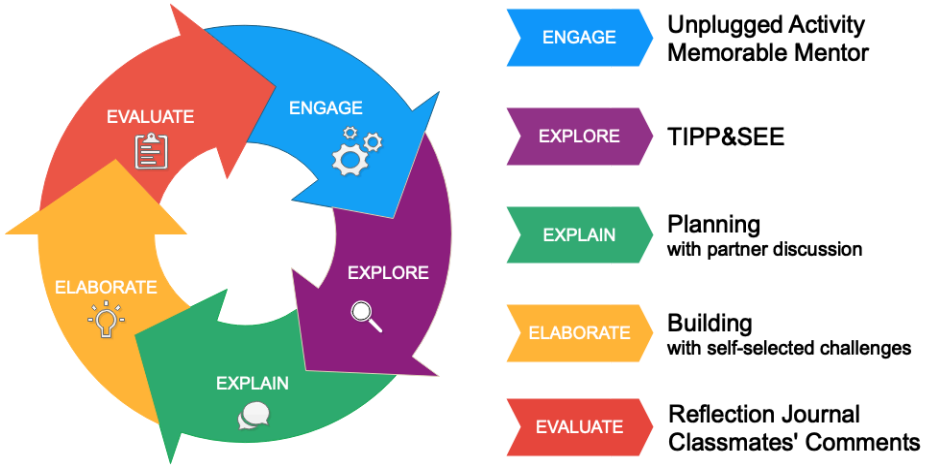
TIPP&SEE

This is a learning strategy that guides students through an existing Scratch project and teaches them to identify what a project does and the code that makes it work. The TIPP&SEE activities target the Scratch-based learning objectives as well as the CS conceptual objectives for each unit. When applying this strategy in the classroom, it is essential for the teacher to model the strategy (I DO), then have students do it together with the teacher (WE DO), so that students can do it themselves (YOU DO).



5e Instructional Model

This model reflects the constructivist theory of learning, where students construct knowledge and understanding through experiences. It is an inquiry-based approach that is grounded in active learning, as students are engaged in the topic, motivated to learn, and guided toward developing the target skills. The five phrases in the 5e model are Engage, Explore, Explain, Elaborate, and Evaluate. Each of the 4 units in the ECforALL curriculum guides students through these five phases.



Culturally Responsive Pedagogy

Culturally Responsive Pedagogy (CRP) is a pedagogical strategy that aims to engage culturally and linguistically diverse students in the classroom and to validate and to empower them . There are numerous ways that CRP has been implemented into classrooms. Five commonly described aspects of culturally responsive instruction include: (1) helping students to value the perspectives of others, (2) affirming a sense

of belonging, (3) validating students' cultural identities, (4) educating students about the diversity of the world around them, (5) supporting students to develop an awareness the impactful social structures. Research has shown repeatedly that CRP increases the academic success of ethnically diverse communities. As educators strive to broaden the participation of students from underrepresented communities in the field of computer science, we believe that the integration of CRP principles into computational thinking curricular materials as well as into professional development for teaching those materials will promote student success and increase interest in the computer science field. We will share learning activities that involve the cultural content, diverse role models in the computer science field, student reflection, and discussion.

5Ws - English Language Arts (ELA)

The 5Ws (**W**ho, **W**hat, **W**hen, **W**here, and **W**hy) scaffold is used for both reading comprehension and pre-writing in classrooms to identify story elements. We use this strategy in the curriculum to help students visualize and plan their Scratch "Stories" before coding. The create project planning sheets contain the 5W questions: **Who** are the sprites (characters)? **What** will the sprites be doing/actions (plot)? **Where** does the project take place (backdrop/setting)? **When** will the actions take place (user events)? The **Why** question was intentionally left out, as our research indicated that students did not understand how to implement this into their projects. Our intent is for students to use something familiar (low cognitive load) to support their novice coding (higher cognitive load) efforts. This approach still encourages and supports the individual creativity and flexibility central to the design of Scratch.

Curriculum Design Overview

CONTENT of the CURRICULUM

The ACT 1 ECforALL Curriculum consists of 19 50-minute lessons, grouped into 4 units: Introduction to Scratch, Sequences, Events, and Loops. The ACT 2 curriculum consists of 20 50-min lessons, grouped into 4 units: Review/Animation, Loops with Conditions, Synchronization, and Variables. The structure of each unit follows the 5e Model (Engage, Explore, Explain, Elaborate, Evaluate), mentioned above.

ACT 1	ACT 2
Unit 0 Setting Up Scratch (1 Lesson)	Unit 0 Setting up Scratch (1 Lesson)
Unit 1 Scratch Basics (4 Lessons)	Unit 1 Review/Animation (5 Lessons)
Unit 2 Sequences (5 Lessons)	Unit 2 Loops with Conditions (5 Lessons)
Unit 3 Events (5 Lessons)	Unit 3 Synchronization (5 Lessons)
Unit 4 Loops (5 Lessons)	Unit 4 Variables (5 Lessons)

The five lessons in Units 2-4 follow the same sequence of learning activities. Each learning activity is described below.

a. **Intro to Concepts**
   The aim of this section is to introduce the unit's key concept(s) in a couple different ways. The first way is to make a connection between the students' everyday lives and the target concept.

b. **Unplugged**
   The aim of this activity is to provide an example of the target concept in a non-digital context. This type of activity appeals greatly to kinesthetic learners, but it is a fun, game-like way to learn and practice the unit's key concept.

c. **Memorable Role Model**
   The aim of this activity is to introduce students to people from diverse backgrounds who have made an impact on the world through computer science, to think about what we can learn from other people, to illustrate a quality that often accompanies success in the field of computer science, to encourage students to develop a sense of identity as a computer scientist by seeing themselves in the mentors that are introduced, and, lastly, to interact with these ideas through listening, reading, writing, and speaking. The curriculum does not explicitly draw parallels between the role models and the students. Please be sure to do so.

d. **Exploring a CS concept**
   The aim of this section is to go a little deeper exploring the unit's target concept. There are exercises that students work through to identify if students are understanding, or not. The class works through these exercises together.

e. **TIPP&SEE**
   The aim of this activity is to guide students through an existing Scratch project and to teach them to identify what a project does and the code that makes it work. When students get to the Quest section, please remind them to remix the project, share it, and add it to the studio you create for it. In order to promote partner work and collaboration, considering putting students in pairs when they

work on their Quests. The Quests should be challenging for students; it's better for students to complete one challenge well than to try superficially at all of them.

f. Plan

In this section students plan their projects. In Unit 4, students plan and build their projects in pairs. Take some time to think about how you will guide and model your students in working well together. Sentence frames are provided on the slides to help students communicate effectively and appropriately with each other. The checklist (or rubric) is presented in this lesson. Please take the time to discuss it in detail with your students, particularly the first column. The rubric offers differentiation for students. The column on the right describes the standard criteria. Students can challenge themselves to meet the criteria in the second or third column on one or more of the rows. When they do, encourage them to problem solve and independently figure out how to complete those more advanced tasks.

g. Build

In this section students build their projects. This section tends to take the longest, as students are developing their projects and thinking of more and more ways to improve it. Please be clear about how many minutes you will give students to develop their project and let them know there will be stopping point. Having deadlines is something that you can emphasize as a real part of a computer scientist's job.

NOTE:

- Be sure students are working on the basics that meet the baseline criteria BEFORE they tinker with backgrounds and sounds.
- Links to tutorials for the "Things to Try" section on the rubric are NOT provided to students in order to encourage their resourcefulness. Teachers, however, have links in the "Online Resources" page.

h. Vocabulary Review

The vocabulary review page in the student workbook gives an opportunity for students to express the unit's key concept in words or in illustration. Teachers can choose to review verbally as a class for the sake of time.

i. Reflection Journal

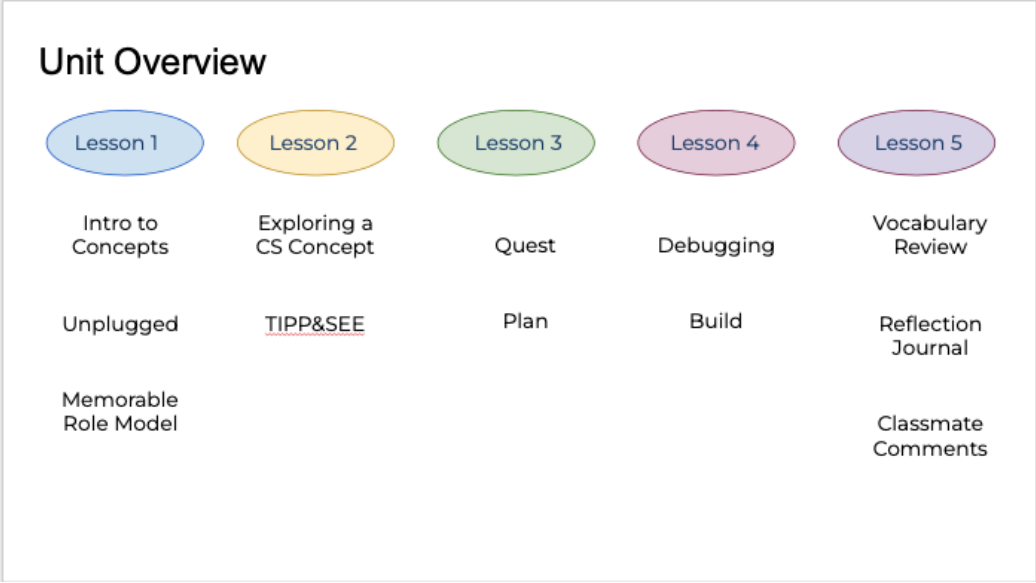
The aim of this section is to guide students to reflect on their experience of doing the CREATE project in each unit. Sentence frames are provided for each question to help students articulate their thoughts. There is a reflection journal at the end of all but the first unit. The first question of each reflection journal asks students to explain something about their project using the target computer science vocabulary emphasized in the unit. The second question in units 2 through 4 guide the students to reflect on an aspect of the creation process. The last two questions of each reflection journal are the same in each unit. One question asks students about a problem that they faced and what they did to respond to it, and the other asks students what they did or what they learned that they are proud of.

We strongly suggest that you read each question to the class and provide time for students to answer each question before you move on to the next question. It is often helpful for students to share their ideas with the class before they write down their answers to help students begin to brainstorm. Please emphasize to students that they are NOT required to use the sentence frames when answering the questions. They are only there to support them if they are stuck.

If you have extra time, it may be fun for students to share their reflections in the interactive Inside Outside Conversational Circles.

j. Classmate Comments

The aim of this section is three-fold: (1) to give students the opportunity to have an authentic audience for their project, (2) to receive (hopefully helpful) feedback from their classmates in their workbooks to have to review again later, and (3) to get ideas from the projects their classmates are producing. Students will need to be able to find their classmates' projects in your Scratch Studio, so it is important that projects are labeled in a systematic way, such as the name and last initial of the student and then the name of the project. The students exchange workbooks and then find their classmate's project in your Scratch Studio. The classmate will fill out the "A Star and a Wish" table: something they liked about the project and one way to make it even better. This process requires modeling to encourage students to provide quality comments. Please project the slide with sentence frames to support students in providing feedback. An alternative to having students give feedback through the "A Star and a Wish" table is to conduct a Gallery Walk. See the link for details.



NOTE: Lesson 5 of each unit has a vocabulary, reflection journal, and classmate comments learning activity. Most teachers find it cognitively draining as well as too time-consuming to complete all three learning activities in each unit. We suggest that teachers choose one or two activities per unit that would be most beneficial to students, alternating the activities each unit.

Components of the Curriculum

The curriculum was designed to support you in engaging your students in computer science, giving them CS skills, and helping them to develop an identity as someone who can do computer science. There are 4 components to help you do so: the Teacher's Guide, Lesson Slide Decks, Student Workbook, and a Lesson Resource Page.

The **Teacher's Guide (TG)** provides background for each unit and the lesson plans with images of each lesson's slides and links to all material you and your students will need. The background information for each unit will include:

- an overview of the unit
- two sets of student learning objectives (CS Learning Objectives and Scratch-based Learning Objectives)
- the unit's vocabulary that will be taught explicitly
- a blue box with the main concept that should be conveyed throughout the unit
- a unit walk-through that gives you a synopsis of the aims of each lesson
- a link to a page with all online resources that will be used in the unit. (These resources are also provided within the lesson plans and the Student Workbook.)

Each lesson plan will provide the student learning objectives, a Learning Activity Summary with suggested times for each activity, the materials both students and teachers will need for the lesson, and then the lesson presentation, which includes images of the lesson slides with notes on important points to mention.

Icons in the Lesson Plan Notes

1.  This is a icon in the Learning Activity Summary in most lessons that signals a good stopping point if you choose to break the 50-minute lesson into two 25-minute lessons. Some teachers prefer to teach this curriculum twice a week for a shorter amount of time in order to fit it into their schedules and to provide more frequency for students. There is an "Optional: Turn and Talk" prompt after each of these symbols and a slide in the Lesson Slide Deck that can be used to refocus students if you choose to teach the two lessons. NOTE: Because the Turn and Talk activities are optional, sometimes they will be the last slide in the lesson slide deck so it does not distract from the flow of the lesson.

2.  This icon and the colored box represents information that you can explicitly tell your students when presenting a particular slide.

3. "*in Student Workbook"* will appear above each slide that is a duplication of a page in the Student Workbook. We included this to help you to support the students in reinforcing instructions and concepts you are conveying.
4. This icon guides teachers to locate the instructions and files needed if they choose to pursue an activity for online instruction.

Lesson Slide Decks (SD) help you to walk the student through the sequence of each lesson. The link to the SD are found in the Teacher Materials section of the TG. Many of the slides are screen shots of the Student Workbook to provide continuity and clarity for students as the lesson progresses.

Lesson Slide Deck Notes:

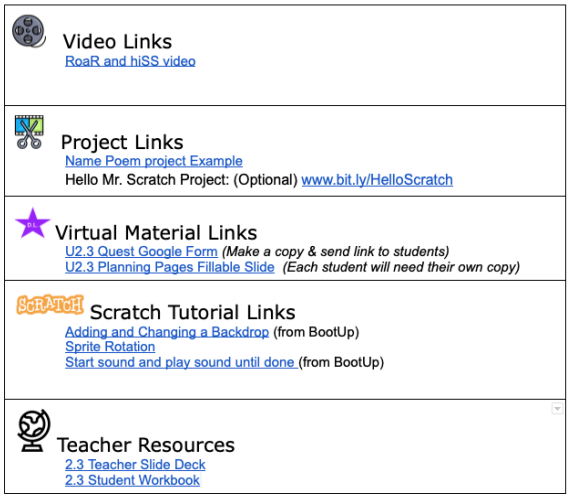
1. Please be sure to preview the slides before you present them. Some contain audio that you should be aware of. Others contain animations, such as the revealing of answers to questions, that are helpful to be apprised of.

2. If a slide represents information in the Student Workbook, the following symbol will be at the bottom right-hand corner of the slide: 

Student Workbook (WB) is designed to solidify the learning of the computer science concepts by visually presenting them to students, by providing clear instructions for the activities, and by giving students a place to demonstrate their understanding of the target concepts, to record their project plans, to self reflect, and to receive feedback from their classmates. Each unit has been given a different color to help you and the students find the right section more easily.

The **Lesson Resource Page** provides quick links to all online resources needed in a given lesson. Each resource page provides links to lesson videos, projects, virtual material, Scratch tutorials, and and resources to support the teacher.

EXAMPLE: **Lesson 2.3 Resources**



Virtual Instruction

While the ECforALL curriculum was designed for an in-person, teacher-student environment, it can be adapted for remote instruction. The table below displays what topics are appropriate for synchronous or

asynchronous learning. Topics that are suggested as asynchronous activities can be assigned with little to no instruction required. The student worksheets associated with these asynchronous topics were converted into virtual files for your students to complete remotely. Topics suggested for synchronous learning are those that require more group discussion or collaboration for a full learning experience.

Italicized activities are those that you can skip due to the limited time you have with your students and/or the need for minimal use of video call features (e.g. screen share, chat). These italicized activities will also be listed in the Learning Activity Summary which is located at the beginning of each lesson. Some synchronous activities would require students to be comfortable with navigating the virtual classroom environment (i.e. zoom, google meet) and their browser (i.e. having multiple tables open).

All topics require access to the Student Workbook as a guide or reference to complete these activities for its respective unit.

Overall Unit Structure	**Asynchronous** Links to virtual file below (*Google Forms, **Fillable Google slide, and/or ***Scratch)
Lesson 1 Intro to Concept	• **Memorable Role Model worksheet* Unit 1 || Unit 2 || Unit 3 || Unit 4
Lesson 2 TIPP & SEE	• *TIPP & SEE worksheet Unit 2 || Unit 3 || Unit 4
Lesson 3 Project Plan	• **Project planning worksheet Unit 2 || Unit 3 || Unit 4 || Unit 5
Lesson 4 Project Build	• *QUEST worksheet Unit 1 || Unit 2 || Unit 3 || Unit 4 • ****Debugging Challenge* ○ Unit 2: bit.ly/debugname ○ Unit 3: bit.ly/aboutbug
Lesson 5 Project Reflect & Share	• **Vocabulary Worksheet Unit 2 || Unit 4 || Unit 5 • *Reflection Journal Worksheet Unit 2 || Unit 3 || Unit 4 || Unit 5 • **Feedback Classmate Comments Unit 2 || Unit 3 || Unit 4 | Unit 5

*Google Forms: The student worksheet for some activities were converted to Google Forms that your students can submit for completion.

**Fillable Sheet: Some student worksheets require students draw or sketch their ideas, therefore these activities were converted into fillable sheets.

***Scratch: These activities would require students to submit their Scratch projects via the Scratch Class Studio or Google Classroom.

Synchronous Instruction

Conducting some activities synchronously will require students to be comfortable using the following features of a video call platform:

- Share screen and stop share screen
- Use the chat box to access links
- Enter a separate virtual room and returning to the main virtual room
- Navigate between two to three tabs

Synchronous Distance Learning Instruction for specific activities

We provide some instructions on how to conduct portions of the curriculum synchronously. Use these purple Distance Learning star icons to locate the instructions and files you need if you choose to pursue an activity for synchronous instruction. ***Remember that italicized activities are recommended to be skipped if synchronous time with your students are limited and/or if you prefer to use minimal video call features.***

CSTA, ISTE, and CCSS ELD/ELA Standards Alignment

This curriculum is aligned with international ISTE and National CSTA Computer Science Standards as well as with the California ELA (English Language Arts) and ELD (English Language Development) state standards. The alignment of these standards for the entire curriculum can be found in Appendix A. The alignment of these standards for each individual unit can be found on the first page of each unit below the student learning objectives.

	CSTA Standards Grades 3-5									
AB-AP-08	Compare and refine multiple algorithms for the same task and determine which is the most appropriate. (P6.3, 3.3)				x		x			
AB-AP-08	Create programs that use variables to store and modify data. (5.2)								x	x
1B-AP-10	Create programs that include sequences, events, loops, and conditionals. (P5.2)	x	x	x	x	x	x	x	x	x
1B-AP-11	Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. (P3.2)		x	x	x	x	x	x	x	x
1B-AP-12	Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. (P5.1)	x	x	x	x	x	x	x	x	x
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)	x	x	x	x	x	x	x	x	x
1B-AP-16	Taking on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. (P2.2)				x					x
1B-AP-17	Describe choices made during program developmint using code comments, presentations, and demonstrations. (P7.2)	x	x	x	x	x	x	x	x	x
IB-IC-21	Use public doman or creative commons media, and refrain from copying or using material created by others without permission. (P7.3)		x	x	x	x	x	x	x	x
1B-AP-15	Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. (P6.1, P6.2)	x	x	x	x	x	x	x	x	x

	California Computer Science Standards									
	Grades 3-5									
3-5.AP.10	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.				x		x			
3-5.AP.11	Create programs that use variables to store and modify data. (P5.2)									x
3-5.AP.12	3-5.AP.12 Create programs that include events, loops, and conditionals. (P5.2)		x	x	x	x	x	x	x	x
3-5.AP.13	Decompose problems into smaller, manageable tasks which may themselves be decomposed. (P3.2)	x	x	x	x	x	x	x	x	x
3-5.AP.14	Create programs by incorporating smaller portions of existing programs, to develop something new or add more advanced features. (P4.2, P5.3)		x	x	x	x	x	x	x	x
3-5 AP.17	Test and debug a program or algorithm to ensure it accomplishes the intended task.	x	x	x	x	x	x	x	x	x
3-5 AP.18	Perform different roles when collaborating with peers during the design, implementation, and review stages of program development.				x					x
3-5 AP.19	Describe choices made during tprogram development using code comments, presentations, and demonstrations.	x	x	x	x	x	x	x	x	x

Common Core State Standards for ELD and ELA												
		Intro to Concepts	Unplugged	Memorable Role Model	Exploring a CS Concept	TIPP&SEE	Pair Program	Plan	Build	Reflection Journal	Classmate Comments	Turn and Talk
4.1.1 (Corresponding CCSS for ELA: SL.4.1, 6; L.4.1, 3, 6)	Exchanging information and ideas with others through oral collaborative discussions on a range of social and academic topics											
4.1.2 (Corresponding CCSS for ELA: SL.4.1, 6; L.4.1, 3, 6 W.4.6; L.4.1, 3, 6)	Interacting with others in written English in various communicative forms (print, communicative technology, and multimedia)											
4.1.3 (Corresponding CCSS for ELA: SL.4.1, 6; L.4.1, 3, 6)	Offering and supporting opinions and negotiating with others in communicative exchanges											
4.1.4 (Corresponding CCSS for ELA: W.4.4–5; SL.4.1, 6; L.4.1, 3, 6)	Adapting language choices to various contexts (based on task, purpose, audience, and text type)											
4.1.5 (Corresponding CCSS for ELA:SL.4.1–3; L.4.3)	Listening actively to spoken English in a range of social and academic contexts											
4.1.11 (Corresponding CCSS for ELA: W.4.1, 4, 9–10; SL.4.4, 6; L.4.1–3, 6)	Supporting own opinions and evaluating others' opinions in speaking and writing											
4.1.12 (Corresponding CCSS for ELA: W.4.4–5; SL.4.4, 6; L.4.1, 3, 5–6)	Selecting and applying varied and precise vocabulary and other language resources to effectively convey ideas											

	ISTE Standards									
Empowered Learner	Students leverage technology to take an active role in choosing, achieving and demonstrating competency in their learning goals, informed by the learning sciences. Students:									
1a	1a. articulate and set personal learning goals, develop strategies leveraging technology to achieve them and reflect on the learning process itself to improve learning outcomes.		x	x	x	x	x	x	x	x
1d	1d. understand the fundamental concepts of technology operations, demonstrate the ability to choose, use and troubleshoot current technologies and are able to transfer their knowledge to explore emerging technologies.	x	x	x	x	x	x	x	x	x
Digital Citizen	Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical. Students:									
2b	engage in positive, safe, legal and ethical behavior when using technology, including social interactions online or when using networked devices.	x	x	x	x	x	x	x	x	x
Innovative Designer	Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions. Students:									
4a	know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or solving authentic problems.	x	x	x	x	x	x	x	x	x
4d	exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.	x	x	x	x	x	x	x	x	x
Creative Communicator	Students communicate clearly and express themselves creatively for a variety of purposes using the platforms, tools, styles, formats and digital media appropriate to their goals. Students:									
6b	create original works or responsibly repurpose or remix digital resources into new creations.	x	x	x	x	x	x	x	x	x