Microsoft

TEALS summer training asset sheet

This sheet will serve as your guide for summer training. It contains relevant links to resources you need through the synchronous sessions. For each synchronous session there is section on pre-work, a place to take notes, the topics for discussions in the synchronous session and the homework to be completed before the next synchronous session.

You will need to **download this document locally** for your notes to be saved.

Table of contents

Meet and greet!

This is an opportunity for teams who have at least one new teaching team member to build team trust and get to know each other and understand their classroom demographics. If you need help locating your school's demographic information visit the links below.

- Search for US Public Schools (ed.gov)
- Search for US Private Schools (ed.gov)
- Search for Canadian Schools

Meet and greet!: Pre-work

- **New classroom teachers and new volunteers:** Complete the "Welcome to TEALS" course in eLearning.

Meet and greet!: My notes

Meet and greet!: Notes and discussion guidance

Topic	Notes/discussion guides
In your notes	**How do you think your background is like your students'?** **How do you think your background is different from your students'?**
Team Reflections	**After introducing yourselves and sharing contact info:** Discuss… • How are the cultural and academic backgrounds of your team members like your students' experience? • How are the cultural and academic backgrounds of your team members different from your students' experience? Consider… • The answers each team member added to the charts • Your school's student demographic data

Meet and greet!: Work to complete by next synchronous session:

New volunteers	New classroom teachers	Returning volunteers	Returning classroom teachers
• Culturally responsive teaching (asynchronous) • Pre-work for Best practices in cs education • Start your volunteer curriculum training • Feel free to begin supporting the CS classroom Part I	• Culturally responsive teaching (asynchronous) • Pre-work for Best practices in cs education • Feel free to begin D&I next steps • Begin learning your curriculum through the TEALS training or your provider training	• None (feel free to start student centered learning) • Start your volunteer curriculum training	• D&I next steps (asynchronous)

Team planning #1

This planning meeting is designed to have teams determine how they will implement initiatives for inclusive learning space and inclusive instruction. This is also a chance to make sure teams have completed certain logistical items in preparation for the school year.

Team planning #1: Pre-work

- Teachers: Complete the “D&I next steps” course in eLearning.

Team planning #1: My notes

Team planning #1: Notes and discussion guidance

Topic	Notes/discussion guides
The classroom plan	Find the link to the classroom plan on your TEALS dashboard
Planning time breakdown	Part 1: Diversity and inclusion planning Part 2: Logistics discussion and checklists by role Part 3: Check-in and dismissal with RM
Diversity and inclusion initiatives checklist	• Teachers: share out the resources you selected from the D&I next steps asynchronous training. • Teaching team: review those resources and decide your initial steps in making your learning space and instruction inclusive.
Logistics and discussion: Teachers' checklist and discussion points	• Claimed Classes • Shared the plan for how classes will begin in fall (hybrid, virtual, in-person) • If hybrid or virtual, what platform are you using? • Are visitors allowed in school? If you don't know do you know when a decision will be made? • Share the finalized classroom enrollment or a timeline for enrollment to be finalized. • Can you provide volunteers with a class roster with student names and pictures for volunteers to study and get familiar with the students?
Logistics and discussion: Volunteers' checklist	• Updated mailing address • Entered your t-shirt size • Signed your volunteer agreement • Received your TEALS t-shirt in the mail yet? • Received your books? (***AP volunteers only***) – AP CS A: eBooks (Building Java Programs and Barron's AP CS A) – AP CSP: eBook (Barron's AP CS P) • Have you started your asynchronous training content? • If needed, have volunteers been set up with school ID/badge, parking pass, and/or a network login?

Topic	Notes/discussion guides
Logistics and discussion: Teaching team checklist	- Do volunteers have information from the school yet regarding onboarding (e.g., background check, etc.)? If yes, have volunteers begun the process? If no, can teachers help get the process started or have the school contact volunteers to start the process? - If using a virtual learning platform, does everyone have access to resources to use if there are technical issues? - Have a shared location for the classroom plan and a timeline for completion. – Link to sample classroom plan

Team planning #1: Work to complete by next synchronous session

New volunteers	New classroom teachers	Returning volunteers	Returning classroom teachers
- Pre-work for best practices in CS education - Continue your Volunteer Curriculum Training - Continue working on Supporting the CS Classroom Part I	- Pre-work for best practices in CS education - Continue learning your curriculum through the TEALS training or your provider training	- Student centered learning (asynchronous) - Start your Volunteer Curriculum Training (*if you have a new curriculum*)	- None

Best practices in CS education

This training is for new teachers and new volunteers to help them understand techniques used specifically for teaching computer science. This training also outlines different roles and responsibilities depending on school structure (virtual vs. in-person).

Pre-work

Roles and responsibilities (On the next two pages, read about roles and responsibilities to prepare for discussion.

Teacher Responsibility

Manage Classroom and School Issues

Goal: Empower volunteers

- Classroom management
- Class administrative tasks (attendance, grades, etc.)
- Share the school calendar
- Set the pace of class
- Coordinate support for students

Lead the teaching team

Goal: Coordinate team members

- Collaborate with team members
- Coordinate the schedule
- Facilitate daily handoffs and weekly syncs
- Set up a channel for quick communication
- Synchronize grading standards

Give and take feedback

Goal: Create an atmosphere where regular feedback among the team is expected

- Be an active listener
- Ask clarifying questions
- Call out what was done well
- Provide constructive feedback
- Coach volunteers as they develop their teaching practice
- Solicit feedback from volunteers on lessons you teach

Learn computer science

Goal: Demonstrate content mastery by progressing through the models

- Participate in content training
- Work through the lessons and homework
- Turn to your volunteers for questions about content
- Model learning CS content
- Take on more responsibilities each year

Table of Contents

Volunteer Responsibility

Classroom Teacher Support

Goal: Create a sustainable CS program

Share content knowledge

Advise teacher in prepping lessons

Attend class consistently

Help with grading, in compliance with policy

Communication

Goal: Actively communicate with team members

During Training:

- Inform the team of your availability
- Make sure your team (and TEALS) have multiple ways to contact you
- Check your primary email regularly
- Provide contact info for time sensitive communications

During the School Year:

- Make sure your team (and TEALS) are informed of any changes in your schedule or availability
- Actively participate in team syncs
- Contribute to the team communication channel

Instruction

Goal: Teach new CS concepts to students

Volunteers in the teacher role:

- Prepping lessons, customizing materials as needed
- Coordinating any formative assessments with teacher
- Teaching prepped lessons

Volunteers in the teaching assistant role:

- Observe all lessons
- Watch for any gaps in instruction or student misconceptions
- Complement lesson with industry relevant experience

Lab Management

Goal: Allow students to practice the content skills they learned

Model how to troubleshoot and debug

Apply strategies from training to encourage students as they create their own programs and projects

Help students find resources to solve their own problems

Check-in with each student at least twice during lab time

Update the team on each student's progress

Best practice in CS education: Notes and discussion guidance

Topic	Notes/discussion guides
Reflections	What roles surprised you? What roles excite you? What roles concern you?
Reflections	**Discuss with your group members the roles/responsibilities** • What roles surprised you? • What roles excite you? • What roles concern you?
Breakout discussion: Building relationship	In your groups take turns sharing out ways you can foster connection with your students.
Breakout activity: Discussing busting bias	**Breakout with your groups and discuss one or two of the following scenarios:** 1. A student asks for more time to complete an assignment. Do you grant the extension? 2. A student is surfing the internet during lab. Do you reprimand them? 3. You're visiting the cafeteria during lunch to recruit students for next year's CS class. Which students should you approach? *How can bias impact decision making in the following scenarios? How can these decisions lead to inequitable outcomes? How can these scenarios become opportunities for more equitable outcomes?*
Activity: Comparing classrooms	Find the links to the 2 classroom photos in your Summer Training Asset sheet Classroom 1 Classroom 2 In your breakout rooms, discuss the differences between the classrooms **Reflect:** Are there particular cultures, attitudes, religions, genders implied in the wall covering? How may those difference appeal to certain students? How may those differences have the opposite effect?

Work to complete by next synchronous session:

New volunteers	New classroom teachers
• Start Pre-work handout for Best Practices in CS Education • Continue working on Supporting the CS Classroom Part I • Continue working on the Volunteer Curriculum Training	• **D&I Next Steps** • Start Pre-work for Best Practices in CS Education • Continue learning your curriculum through the TEALS training or your provider training

Review the following material on:

- Memory diagrams
- The four steps to solving any problem
- Debugging with students
- Bloom's taxonomy in CS

Memory diagrams

- are a visual representation of the state of the computer memory during the execution of a program.
- are typically used while tracing through a code example.
- mimic a "debugger."

Watch the following videos on memory diagrams:

- Ithaca style memory diagram
- Trace table

The four steps to solving any problem

Step	Some questions you can ask
Understand the problem	• What are you trying to find or do? • Can you restate the problem in your own words? • Can you think of a picture or diagram that might help you understand the problem? • Is there enough information for you to find a solution? • Do you understand all the words used in stating the problem? • How will you know you have a correct solution?
Plan a solution	• Can you write out the steps of how to solve the problem in plain English? • Are there any constructs you know that relate to the problem? • Can you break the problem down into smaller pieces? • Are there any patterns? (logic, looping) • Can you think of a similar problem you've solved? • Are there any special cases you need to consider? (e.g., edge cases) • Can you work backwards (bottom-up)? • Can you introduce a supplemental element that will help you?
Implement the solution	• Is each part of your solution implemented? • Does each part of your solution work? (compared to specification) • Is your code organized in easy-to-understand modules with comments? • Did you read your code line-by-line (troubleshooting/debugging)?
Reflect on the solution	• Does your answer make sense? Does it solve the original problem? • What worked and what didn't? • Can you think of another (easier or more efficient) method to solve the problem?

Subgoal labeling

- Models breaking a problem into parts
- Give a short, clear label to each part
- Incorporate the labels into and/all of:
  - Lab/project instructions
  - Starter code template ("code skeleton")
  - Sample solutions

Subgoal labeling in lab instructions

Without labels	With labels
1. Click on "My Blocks" to see the blocks for components you created 2. Click on "clap" 3. Drag out a when clap.Touched block 4. Click on "clapSound" and 5. Drag out call clapSound.Play 6. Connect it after when clap.Touched	**Handle Events from My Blocks** 1. Click on "My Blocks" to see the blocks for components you created 2. Click on "clap" 3. Drag out a when clap.Touched block **Set Output from My Blocks** 1. Click on "clapSound" and 2. Drag out call clapSound.Play 3. Connect it after when clap.Touched

Subgoal labeling in code

Without labels	With labels

`def count_em_all(some_word):`
`count = 0`
`for letters in some_word:`
`count = count + 1`
`return count`
	`def count_em_all(some_word):`
	`# Create a counter`
	`count = 0`
	`# Look at each letter in the word`
	`for letters in some_word:`
	`# Count the letters`
	`count = count + 1`
	`# Return the number of letters`
	`return count`

Debugging with students

Debugging is one of the most common tasks performed by computer scientists

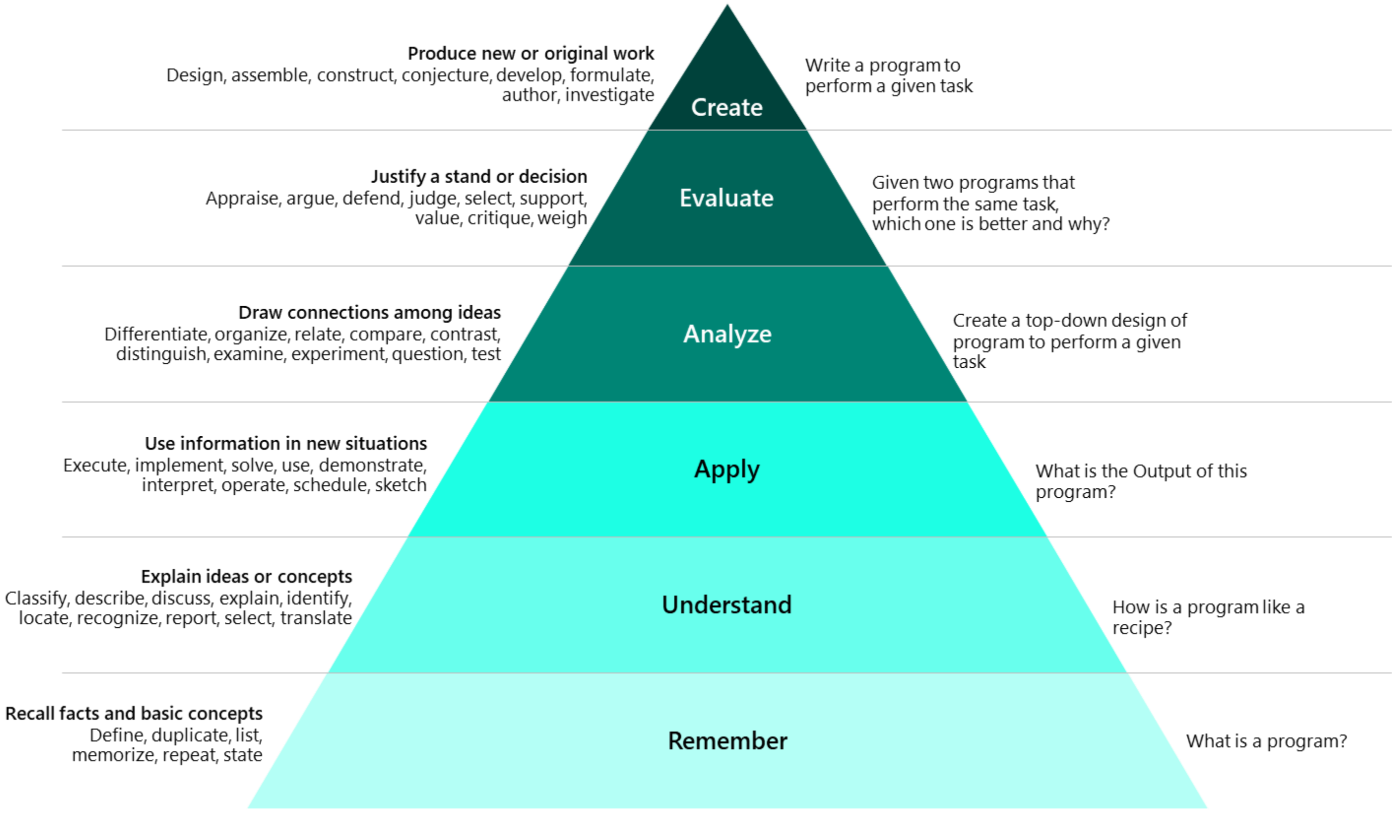
When debugging with students

- Narrate the process: Help students understand your thinking process
- Explain your conclusions: What's obvious to you may not be obvious to students
- Unpack the problem: Students tend to work in smaller chunks

think about…

- What tools and techniques do you use when solving problems?
- How is helping a ***student*** solve a problem different than helping a colleague solve a problem?
- How is helping someone else solve a problem different than solving one for yourself?
- Why might it be important to ***not*** tell a student the solution?

Blooms taxonomy

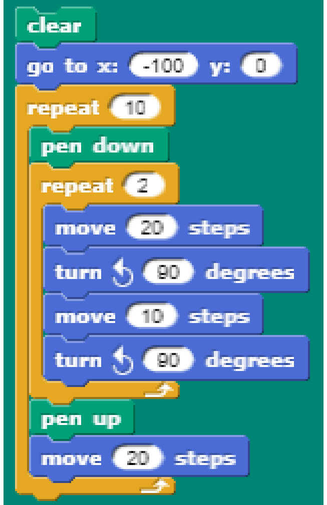

Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
Create
Write a program to perform a given task
Justify a stand or decision
Appraise, argue, defend, judge, select, support, value, critique, weigh
Evaluate
Given two programs that perform the same task, which one is better and why?
Draw connections among ideas
Differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
Analyze
Create a top-down design of program to perform a given task
Use information in new situations
Execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
Apply
What is the Output of this program?
Explain ideas or concepts
Classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
Understand
How is a program like a recipe?
Recall facts and basic concepts
Define, duplicate, list, memorize, repeat, state
Remember
What is a program?

Vanderbilt University Center for Teaching

Best practices in CS education: My notes

Topic | **Notes/discussion guides**

Activity: Code tracing in Snap!



```
clear
go to x: -100 y: 0
repeat 10
    pen down
    repeat 2
        move 20 steps
        turn ↺ 90 degrees
        move 10 steps
        turn ↺ 90 degrees
    pen up
    move 20 steps
```

Trace the first few lines of Snap! code (volunteers should lead any classroom teachers as your group traces the code)

Line 1	
Line 2	
Line 3	
Line 4	
Line 5	
Line 6	
Line 7	
Line 8	
Line 9	
Line 10	
Line 11	
Line 12	
Line 13	

Let’s work through a problem…

Problem: Write a program that prompts the user for a number between 1 and 10. If the number is within range, double its value; otherwise, tell the user that the input was out of range. This is a one time run, only one loop is needed.

In your groups discuss what questions you can ask to verify that a student has completed the following steps to solving the above problem:

1. Understand the problem
2. Plan a solution
3. Implement the solution
4. Reflect on your solution

Refer to your prework for this session.

Activity: Classroom example

During lab you find that:

- 5 students don’t understand loops at all
- 15 are handling the lab just fine
- 5 finished the lab in 10 minutes and are now bored

How can you help those students who are **behind**?
How can you help those students who are **ahead**?
What if a particular student is **frequently** ahead or behind?

<table>
<tr><th>Topic</th><th>Notes/discussion guides</th></tr>
<tr><td>Activity: Offering support</td><td>
<p><b>In your breakout rooms; choose one or two of the given students and their accompanying behaviors.</b></p>
<p>What can you say or do to support the students exhibiting these behaviors?</p>
<p><b>Focus on:</b></p>
<ul><li>How the student is struggling with failure</li><li>How a volunteer/teacher can help</li></ul>
<table>
<tr><td><b>Student 1</b></td><td><ul><li>Frustration</li><li>Not seeking help</li><li>Cheating</li><li>Arguing over grades</li><li>Showing off</li></ul></td></tr>
<tr><td><b>Student 2</b></td><td><ul><li>Shutdown</li><li>Over-reliance on help</li></ul></td></tr>
<tr><td><b>Student 3</b></td><td><ul><li>Paralysis</li><li>Not seeking help</li></ul></td></tr>
</table>
</td></tr>
<tr><td></td><td></td></tr>
</table>

Best practices in CS education: Work to complete by next synchronous session:

New volunteers	New classroom teachers
• **Supporting the CS classroom part I** • **Supporting the CS classroom part I: Lesson planning (co-teach only)** • Supporting the CS classroom part II pre-work • Continue working on your volunteer curriculum training	• Continue learning your curriculum through the TEALS training or your provider training

Supporting the CS classroom part II

This training is for new volunteers and is a continuation of the content from Supporting the CS Classroom part I.

Supporting the CS classroom part II: Pre-work

- New volunteers: Complete the “Supporting the CS classroom Part I” course in eLearning.
- Review: Sample lesson

Supporting the CS classroom part II: My notes

Supporting the CS classroom part II: Notes and discussion guidance

Topic	Notes/discussion guides
Discussion: School norms	In your groups share out at least one question that you have for your classroom teacher about school norms. Take note of any new questions you might ask based on the group discussion.
Activity: Chunking and expert bias	**Refer to the lesson from pre-work. (*TEALS AP CS A Lesson 2.05: for loops*)** 1. Discuss ways you can apply chunking to your lesson 2. Discuss ways to mitigate expert bias when teaching or assisting in this lesson.
Activity: Instructional techniques	**Refer to the lesson from pre-work (*TEALS AP CS A Lesson 2.05: for loops*)** Discuss what instructional techniques can be used to introduce the lesson content.
Activity: Formative assessment	**Refer to the lesson from pre-work. (*TEALS AP CS A Lesson 2.05: for loops*)** 1. Determine at least one place in the lesson where you think there should be a check for understanding. 2. Share one way you can check for understanding at that point in the lesson.
Activity: Lesson review	**Refer to the lesson from pre-work. (*TEALS AP CS A Lesson 2.05: for loops*)** Discuss what questions students might have during the lesson and how you might respond.

Supporting the CS classroom part II: Work to complete by next synchronous session:

New volunteers
• Continue your volunteer curriculum training

Team planning #2

This training is for teaching teams to celebrate completing training, finish their classroom plans and to practice various elements of teaching and facilitating labs.

Team planning #2: Pre-work

- Complete all other required training

Team planning #2: My Notes

Team planning #2: Notes and discussion guidance

Topic	Notes/discussion guides
Teaching practice: Choose your own adventure	Choices: • Co-teaching mock teaching: Practice teaching a sample lesson • Lesson review and preparation: Review a lesson and plan for how your team will support all students in learning the content • Misconception game: Role play getting a student unstuck during a lab assignment • Situation analysis: Review four classroom scenarios and decide how your team would respond
In your break outs…	Complete your classroom plan Complete one of the 'Teaching Practice' Activities

Choose your own adventure

Your team will select one of the following activities

1. Mock teaching

   Practice teaching a sample lesson (Page 2)

   Who should do mock teaching?

   New teaching teams who have been asked to prepare a lesson to teach. If your regional manager did not ask you to prepare a lesson, you should select another activity.

2. Lesson analysis

   Review a lesson and plan for how your team will support all students in learning the content

   Who should do lesson analysis?

   Co -Teach and Lab Support teams teaching TEALS Intro to CS or TEALS AP CS A.

3. Misconception game

   Role play getting a student unstuck during a lab assignment

   Who should do the misconception game?

   Any Co-Teach or Lab Support teams

4. Situation analysis

   Review four classroom scenarios and decide how your team would respond

   Who should do situation analysis?

   Any Co-Teach or Lab Support teams

Mock Teaching

1. Select one volunteer teacher and volunteer TA to present the lecture portion of a lesson.
2. The rest of the team will act as students for the lesson, pretending they have not seen the content before.
3. Those acting as students will also take notes using the Mock Teaching Feedback Form.
4. After 10 minutes, exchange feedback, and then select a new teacher and TA.

Mock teaching feedback form

As your partner team teaches their lesson, take notes using the form below to give them feedback on their work. Remember that for many of you this will be your first time teaching. Focus your feedback on missed opportunities or possible techniques/strategies rather than on small and overall insignificant issues.

Teachers:	**Lesson:**
Used whiteboard effectively? Notes:	*Yes* □ *No* □
Asked questions effectively? Notes:	*Yes* □ *No* □
Incorporated TA/co-teacher/classroom teacher effectively? Notes:	*Yes* □ *No* □
Encouraged and supported students, even when incorrect? Notes:	*Yes* □ *No* □
Instructional/questioning techniques used (including adaptations to the lesson plan) Notes:	
Other items: □ Addressed students by name □ Used inviting and positive tone/body language □ Gave clear instructions	□ Identified learning objectives □ Included a hook in lesson □ Used virtual classroom tools effectively
Overall feedback:	

Lesson analysis

Lesson analysis has two parts

1. **Identify what skills are being taught and how you are going to teach them.**
   a. Computer Science Skills
   b. Other Skills (social Emotional, reading, writing, math)
   c. What instructional strategies will you use to support students learning this material? What roles will the TA, Teacher, and classroom teacher play in making sure students learn this material?
2. **Identify what skills are needed to complete the task and how you are going to support students who haven't mastered these skills.**
   a. Computer Science Skills
   b. Other Skills (Reading, writing, listening, speaking, collaboration, math)
   c. What modifications can you make to the lecture or lab to support students who struggle with these skills? What roles will the TA, Teacher, and classroom teacher play in making sure students learn this material?

You can select any lesson for analysis. If you need assistance, these TEALS lessons lend themselves particularly well to analysis.

AP CS A 1.05

Intro to SNAP! 1.5

Intro to Python 1.02

Sample lesson analysis: SNAP Lesson 2.3

1. **Identify what skills are being taught and how you are going to teach them**
    a. Computer Science Skills:
        i. Inputs
        ii. Conditionals
    b. Other Skills: None
    c. What instructional strategies will you use to support students learning this material?
        i. Lecture –
            1. VT gives the lecture
                a. Socratic Questioning to tie new content to old content
                b. Think-pair-share before cold calling
            2. VTA:
                a. reiterates important points
                b. answers student questions in the chat
            3. Classroom Teacher:
                a. watches for expert bias
                b. clarifies any confusing points.
        ii. Demo
            1. VT walks through the demo
                a. Programs examples from students' lives
            2. VTA:
                a. reiterates important points
                b. answers student questions in the chat
            3. Classroom Teacher:
                a. Monitors student participation
        iii. Lab
            1. VT and TA
                a. Asks students to walk them through their code
                b. Asks about next steps
            2. Classroom Teacher:
                a. Completes lab with students
                b. Monitors student behavior
2. **Identify what skills are needed to complete the task and how you are going to support students who have not mastered these skills**
    a. Computer Science Skills:
        i. Basic scripting in SNAP
            1. VTA is going to run a small group that will code this lab as a group
        ii. Boolean Logic
            1. VT to reiterate the concept during the lecture
    b. Other Skills:
        iii. Basic English reading and writing skills
            2. Read the text on the blocks
                a. Provide starter code that only includes the needed blocks and a visual block guide
            3. Fill out the planning worksheet
                a. Allow students to write the planning sheet in another language
                b. Allow students to draw out the planning sheet.

Misconception game: Round 1

Designate one person to be a student struggling with an assignment and one person to be a TA attempting to help. The TA will only have a few minutes to unblock the student.

TA: Don't tell the student the answer and don't worry about getting completely resolved. Focus on giving the student a clear next step to try.

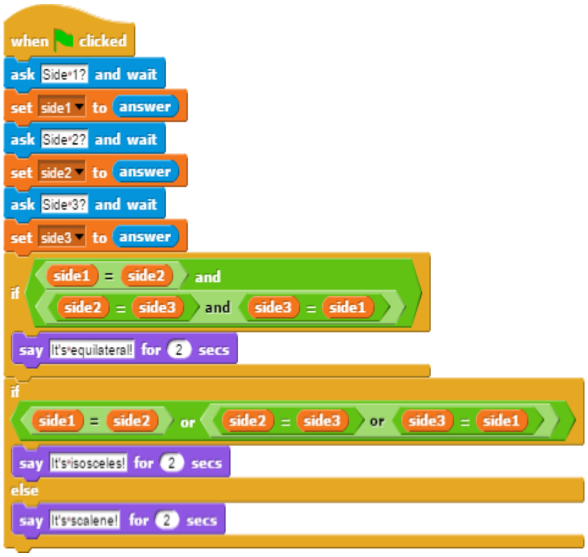
Student: See direction for the scene under each scenario.

<table>
<tr><th></th><th>Instructions</th><th>Student code</th><th>Student code output</th></tr>
<tr>
<td>TEALS Intro</td>
<td>

Write a Snap! script that asks the user for the lengths of all three sides of a triangle. Store each length in a variable. Then, say which type of triangle has the given lengths. Your program should only give one type per set of inputs. A triangle can be one of three types:

<table>
<tr><th>Triangle type</th><th>Description</th></tr>
<tr><td>Equilateral</td><td>All three side lengths equal</td></tr>
<tr><td>Isosceles</td><td>Two side lengths equal, one different</td></tr>
<tr><td>Scalene</td><td>All three side lengths different</td></tr>
</table>

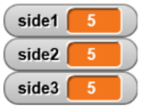
</td>
<td>



```
when [flag] clicked
ask [Side 1?] and wait
set [side1] to (answer)
ask [Side 2?] and wait
set [side2] to (answer)
ask [Side 3?] and wait
set [side3] to (answer)
if <<(side1) = (side2)> and <<(side2) = (side3)> and <(side3) = (side1)>>>
    say [It's equilateral!] for (2) secs
if <<(side1) = (side2)> or <<(side2) = (side3)> or <(side3) = (side1)>>>
    say [It's isosceles!] for (2) secs
else
    say [It's scalene!] for (2) secs
```

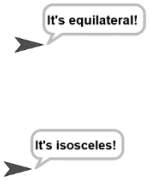
</td>
<td>

Input:



```
side1 5
side2 5
side3 5
```

Output:



```
It's equilateral!
It's isosceles!
```

</td>
</tr>
</table>

Student thinking (only student reads):

Intro: I'm checking whether all 3 sides are equal and then I'm checking whether 2 sides are equal. I don't know why it's saying Isosceles when the triangle is Equilateral.

Student mistake (only read afterwards):

Intro: Student is using if instead of an if-else to check the first condition so the program will say "isosceles" any time two sides are equal (even if all three are equal).

	Instructions	Student code	Student code output

TEALS AP CS A	Write for loops to produce the following output:
	*
	**
	***
	****
	*****
		for (int I = 0; I < 5; i++) {
		for (int j = 0; j < i; j++) {
		System.out.print("*");
		}
		System.out.println();
		}
			*
			**
			***
			****

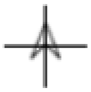
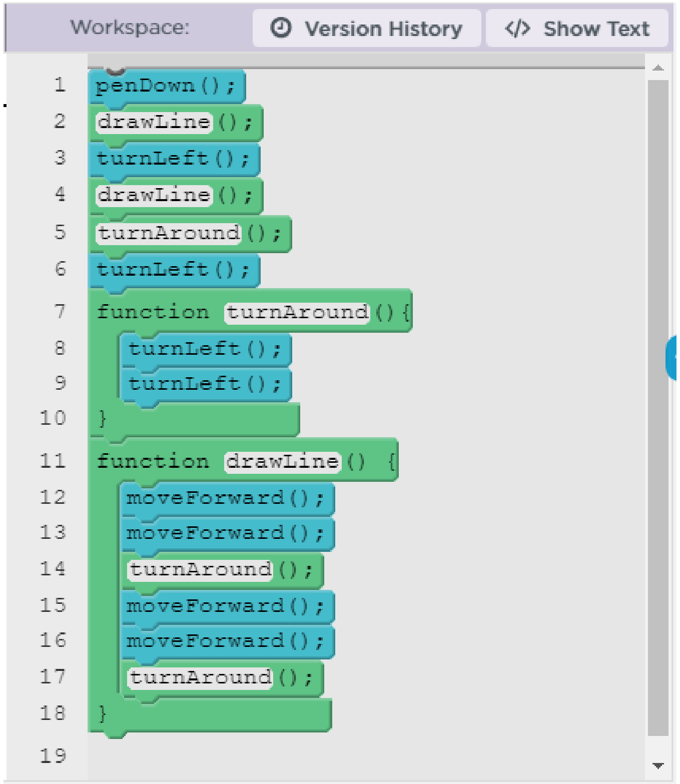
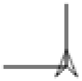
Student thinking (only student reads):

AP CS A: The number of stars in the inner loop is supposed to be dependent on which row number I'm on, so my condition is j < I. I should have that number of stars in each row.

Student mistake (only read afterwards):

AP CS A: The bounds of the inner for loop are off by because the outer loop starts its variable at 0, resulting in one too few stars being printed on each line.

	Instructions	Student code	Student code output

AP CS Principles	Get some more practice calling a function by using turnAround() to create a plus sign, as shown below. We have defined turnAround() for you
	
		
		
		penDown();
		drawLine();
		turnLeft();
		drawLine();
		turnAround();
		turnLeft();
		function turnAround() {
		turnLeft();
		turnLeft();
		}
		function drawLine() {
		moveForward();
		moveForward();
		turnAround();
		moveForward();
		moveForward();
		turnAround();
		}
			

Student thinking (**only student reads**):

AP CSP: A plus is drawn as two line segments 90 degrees apart from each other.

Student mistake (**only read afterwards**):

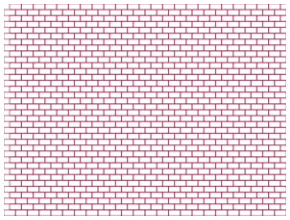
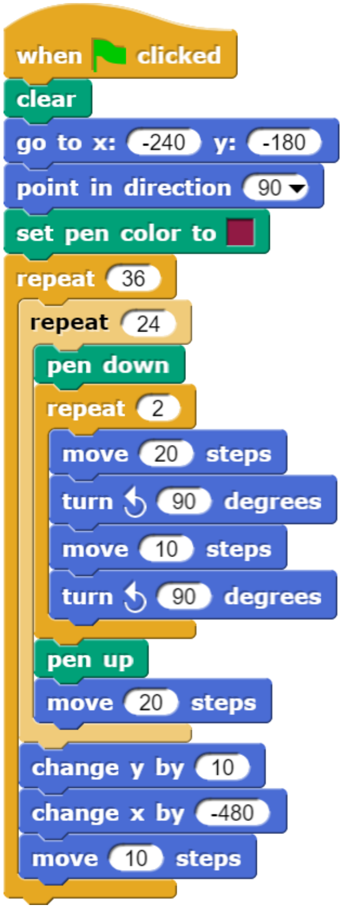
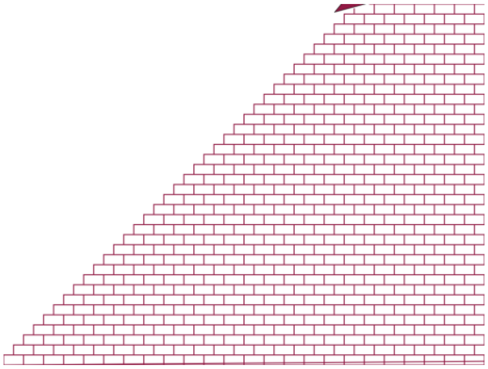
AP CSP: In order to draw a plus with only two lines, the turtle must start and end in the middle of a line segment. As drawLine is defined here, the turtle is starting and ending at the end of a line segment.

Misconception game: Round 2

Switch roles or choose two new people on your team. Designate one person to be a student struggling with an assignment and one person to be a TA attempting to help. The TA will only have a few minutes to unblock the student.

TA: Don’t tell the student the answer and don’t worry about getting completely resolved. Focus on giving the student a clear next step to try.

Student: See direction for the scene under each scenario.

<table>
<tr><th></th><th>Instructions</th><th>Student Code</th><th>Student Code Output</th></tr>
<tr>
<td>TEALS Intro <i>(Lab 3.2)</i></td>
<td>Write a Snap! script to draw a brick wall that fills the stage, like the one below. Each brick should be 20x10 pixels; the stage is 480 pixels wide by 360 pixels tall.
</td>
<td>
<pre>
when ⚑ clicked
clear
go to x: -240 y: -180
point in direction 90
set pen color to ■
repeat 36
  repeat 24
    pen down
    repeat 2
      move 20 steps
      turn ↺ 90 degrees
      move 10 steps
      turn ↺ 90 degrees
    pen up
    move 20 steps
  change y by 10
  change x by -480
  move 10 steps
</pre></td>
<td></td>
</tr>
</table>

Student thinking (**only student reads**):

Intro: According to the problem, every other row should start “indented” by 10 pixels, which is what I did.

Student mistake (**only read afterwards**):

Intro: The code offsets each row by 10 pixels, but always in the same direction, resulting in the turtle constantly moving to the right.

<table>
<tr><th></th><th>Instructions</th><th>Student code</th><th>Student code output</th></tr>
<tr>
<td>TEALS AP CS A</td>
<td>Write a method named smallestLargest that asks the user to enter numbers, then prints the smallest and largest of all the numbers typed in by the user. You may assume the user enters a valid number greater than 0 for the number of numbers to read.</td>
<td><pre>
public static void smallestLargest() {
  Scanner kb = new Scanner(System.in);
  System.out.print("How many numbers? ");
  int numVals = kb.nextInt();

  int min = 0;
  int max = 0;
  for (int i = 0; i < numVals; i++){
    System.out.print("Number " + (i + 1) + ":
");
    int num = kb.nextInt();
    if (num < min) {
      min = num;
    }
    if (num > max) {
      max = num;
    }
  }
  System.out.println("Smallest = " + min);
  System.out.println("Largest = " + max);
}
</pre></td>
<td><pre>
Input:
    5
    2
    8
    9
    4
    7

Output:

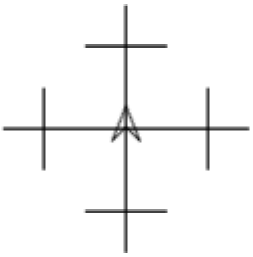
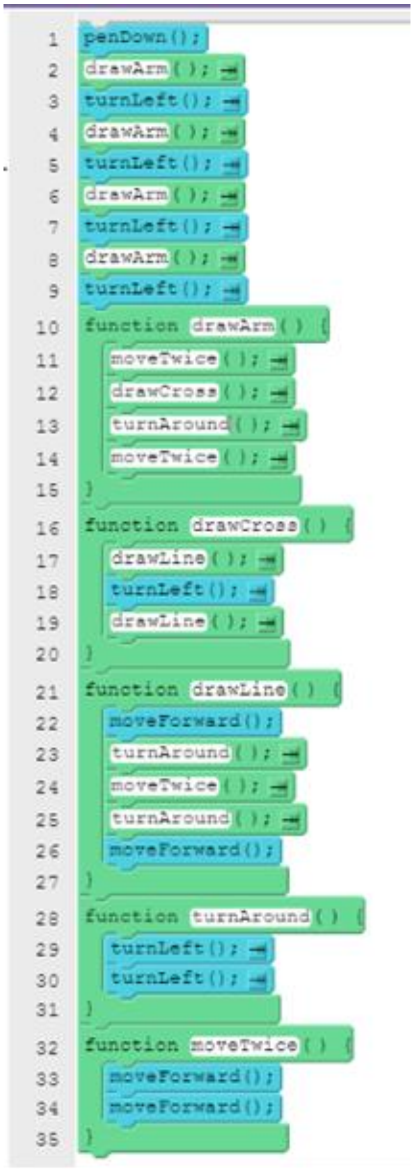
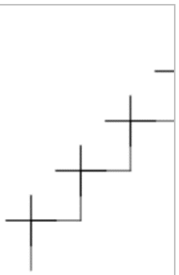
Smallest =
0
    Largest
= 9
</pre></td>
</tr>
</table>

Student thinking (**only student reads**):

AP CS A: I have to initialize the values of variables to something neutral. 0 works for max, so I've used it for min too.

Student mistake (**only read afterwards**):

AP CS A: Since min is initialized to 0, no positive value will ever be considered the minimum. (If negative values are given by the user, the correct minimum will be found, but not the correct maximum.)

	Instructions	Student code	Student code output
AP CS Principles	Write a program to draw the following figure: 		

```
penDown();
drawArm();
turnLeft();
drawArm();
turnLeft();
drawArm();
turnLeft();
drawArm();
turnLeft();
function drawArm() {
  moveTwice();
  drawCross();
  turnAround();
  moveTwice();
}
function drawCross() {
  drawLine();
  turnLeft();
  drawLine();
}
function drawLine() {
  moveForward();
  turnAround();
  moveTwice();
  turnAround();
  moveForward();
}
function turnAround() {
  turnLeft();
  turnLeft();
}
function moveTwice() {
  moveForward();
  moveForward();
}
```

Student thinking (**only student reads**):

AP CSP: drawArm is correctly drawing one arm of the shape and then I'm turning 90 degrees to draw the next one.

Student mistake (**only read afterwards**):

AP CSP: The turtle is turning around after drawing each "cross" instead of turning left. As a result, it does not end up facing the same direction it started after each "cross." (in other words, drawCross has an unintended side effect)

Situation analysis

Read these different scenarios and as a group think about different ways you could manage the scenario.

Scenario 1: The teaching team at South North Western HS are teaching a lesson on how to use a Boolean data type. They base their lesson around an example involving a meal at a restaurant (example: If [food was good] and [service was good] then [give a high tip]). Most of their students seem unengaged.

Why do you think this example didn't engage high school students? Who do you think would be engaged by this question? What could the teaching team have done differently during the lesson?

Scenario 2: Aleesha, the classroom teacher, is leading a lesson on big data. Aleesha teaches the lesson on her own, and when the students begin lab time, there aren't many questions. At the end of the lesson, the volunteers express frustration that they don't feel useful in the classroom.

How could the full teaching team contribute in Aleesha's classroom? What could Aleesha have done differently? How could the volunteer engage with students even if they aren't answering questions?

Scenario 3: Jeremiah frequently has his head down on his desk during instruction, and during lab time he either stares blankly at his screen or plays games. Shanti, one of the volunteers, sees Jeremiah's blank workspace and asks if he needs help. Jeremiah simply responds, "Nah, I'm fine."

How could Shanti and her teaching team engage with Jeremiah? How might the actions of the classroom teacher be different from a volunteer teacher?

Scenario 4: During the Letty's lecture there is a small group of four students who always raise their hands to answer her questions. Other students seem like they are paying attention but are less eager to talk in front of the class. When she checks in with students during the lab they do not have any questions, but the same four students are eager to show the team their code. Everyone else appears to be on task. When the lab is turning it less than 50% of the students can not accurately complete the lab.

When the teaching meets for their weekly meeting, what sort of strategies should they talk about implementing so this won't happen again? Should they reteach the content?

Team planning #2: Work to complete

New volunteers	New classroom teachers	Returning volunteers	Returning classroom teachers
Complete your volunteer curriculum training.	Complete your curriculum training through the TEALS training or your provider training Make sure volunteers are prepared for the start of the school year.	Complete your volunteer curriculum training.	Make sure volunteers are prepared for the start of the school year.