Student Growth Objective Form

(DISTRICT-DEVELOPED SAMPLE SGO for GRADE 10 Computer Science; 1 of 2)

Name	School	Grade	Course/Subject	Number of Students	Interval of Instruction
		10	Computer Science	10	Sept. 2019 – Apr. 2020

Standards, Rationale, and Assessment Method

Rationale

The Introduction to Computer Science curriculum offered through TEALS has a heavy focus on the basic programming components. This curriculum advocates a "hands-on" learning approach in which students' primary means of learning is through discovery, experimentation, and application. To that end, each unit is built around a large, culminating, programming project that exercises the objectives of the unit. The lessons provide the skills and support necessary to enable students to complete the project and demonstrate mastery of the unit's objectives. Substantial class time, in the first semester, focuses on Snap! a free, blocks- and browser-based educational graphical programming language that allows students to create interactive animations, games, stories, and more, while learning about mathematical and computational ideas.

This SGO reflects students' demonstration of mastery of the skills specific to SNAP! And reinforced in each unit as outlined here: https://snap.berkeley.edu/about.html

Grade 10 Standards Addressed within this Student Growth Objective:

UNIT I

- *Define "algorithm"*
- *Construct algorithms for performing simple tasks*
- *Define and identify "blocks," "scripts," "sprites," and "the stage" in SNAP.*
- *Write simple SNAP programs*
- *Describe what simple SNAP programs do without executing the code*
- *Describe what simple SNAP programs do without executing the code*
- *Name the categories of blocks in SNAP and describe what the blocks in each category do*
- *Describe the function of several common SNAP blocks*
- *Be able to use common blocks to build simple SNAP*
- *Construct simple algorithms to draw shapes*
- *Convert algorithms into SNAP programs*
- *Animate SNAP sprites using costume changes and movement*
- *Trigger action in other sprites using broadcasts*
- *Apply basic programming and SNAP skills to create an animated movie, play, nursery rhyme, or other scene*
- *Practice good debugging skills to correct issues as they arise while programming*
- *Explain why loops are useful*

UNIT II

- *Define "loop" in a programming context*
- *Explain why loops are useful*
- *Implement simple repeat and forever loops in SNAP*
- *Utilize loops to reduce redundancy in code*
- *Use nested loops to solve programming problems*
- *Ask for and receive user input in a SNAP program*
- *Use simple conditional (if and if-else) blocks to alter control*

- *Use variables to track values throughout a program*
- *Define and identify Boolean expressions and operators*
- *Evaluate Boolean expressions*
- *Utilize Boolean operators (and/or/not) to create compound conditions*
- *Combine loops with conditionals to create models with repeated but conditional behavior*
- *Implement a well-written version of Pong*
- *Practice good style and conventions to create readable and maintainable code*

UNIT III

- *Define the following terms in a computer science context:*
    - *abstraction*
    - *detail removal*
    - *generalization*
    - *procedural decomposition*
- *Describe how utilizing procedural decomposition can improve the readability and maintainability of algorithms and/or code*
- *Build custom command blocks in SNAP*
- *Recognize opportunities to improve algorithms by abstracting or generalizing parts into sub-procedures*
- *Utilize detail removal and generalization to construct blocks that practice abstraction*
- *Build custom SNAP blocks that take arguments*
- *Build custom reporter and predicate blocks in SNAP*
- *Use loops, variables, and Boolean expressions to implement a Super Mario Bros. style platform game.*
- *Practice good debugging skills to correct issues as they arise while programming.*

UNIT IV

- *Explain the concept of a "list" in a programming context*
- *Identify scenarios in which lists are useful*
- *Create static lists in SNAP*
- *Access elements of a list*
- *Add and remove elements from a list*
- *Traverse a list, accessing each element one at a time*
- *Perform operations combining all elements in a list*
- *Select defined subsets of elements in a list*
- *Traverse a list, accessing each element one at a time*
- *Perform operations combining all elements in a list*
- *Select defined subsets of elements in a list*
- *Explain the sequential search algorithm*
- *Implement several variations of sequential search*
- *Use lists to implement a complete version of "Guess My Word"*
- *Exercise good programming practices to produce code that is functional and well-written*

UNIT V

- *Explain why prototyping and clones can be useful*
- *Describe how complex goals can be accomplished using cloning*
- *Demonstrate the difference between sprite and global variables*
- *Explain how cloning and prototyping simplify working with numerous similar sprites in the same program*
- *Create prototype sprites and clones of the prototype sprite*
- *Explain the difference between a "master" sprite and a "clone" sprite*

- *Pass information from the master to individual clones*
- *Describe a race condition that might occur due using global variables and clones*
- *Delete clones when they are no longer needed*
- *Use cloning to implement a complete version of "Space Invaders"*
- *Exercise good programming practices to produce code that is not only functional but also elegant and well-written*

UNIT VI

- *Identify the key considerations when designing a piece of software*
- *Describe methods for prioritizing features, use cases, and/or scenarios*
- *Explain why design and planning are necessary steps in the software engineering process*
- *Identify factors to use when choosing between project ideas*
- *Rank a group of proposed project ideas using the identified factors*
- *Define key scenarios for a project and the features required to implement each scenario*
- *Explain the importance of wire-framing when designing an application*
- *Identify the main components of a functional project specification and explain the purpose of each section*
- *Develop a project idea into a full, detailed specification*

NJ Computer Science State Standards
Connections Within the 9–12 Framework:

- *Data and Analysis: Storage, Collection*
- *Networks and the Internet*
- *Cyber-security*
- *Network Communication, Coordination, and Organization*
- *System Relationships*
- *Impacts of Computing & Culture*
- *Algorithms and Programming*
- *Variables*
- *Modularity*
- *Abstractions*
- *Computing Systems: Hardware and Software, Troubleshooting*

Assessment Method: The capstone project will be used as a post assessment to measure students' growth.

Starting Points and Preparedness Groupings & Data Measure(s) used to Establish Baseline:

Student tiers will be determined the following:

- Fall *Getting to know you* Lab; no weight
- 2019 PARCC Score for Mathematics; no weight
- 2019 IED End of Course Score; no weight

Each tier will be assigned a target command level.

Preparedness Group		Baseline Score
Low	Tier 1	50 - 59
	Tier 2	60 - 69
Middle	Tier 3	70 - 79
High	Tier 4	80 - 89
	Tier 5	90 - 100

Student Growth Objective

Growth Goal: By April 2020, 70 - 75% of students in each preparedness group will meet or exceed their assigned target command level for full attainment of the objective as shown in the scoring plan as measured by the assigned capstone project.

Preparedness Group (e.g. 1,2,3)	Number of Students in Each Group	Target Command Level
Tier 1		PLD 2
Tier 2		PLD 3
Tier 3		PLD 4
Tier 4		PLD 5
Tier 5		PLD 5*

* Students are expected to maintain a PLD of 5; showing growth within this PLD.

Scoring Plan

State the projected scores for each group and what percentage/number of students will meet this target at each attainment level. Modify the table as needed.

Preparedness Group		Student Target Command Level	Teacher SGO Score Based on Percent of Students Achieving Target Score			
			Exceptional (4) **>75%**	Full (3) **70 – 75%**	Partial (2) **65-69%**	Insufficient (1) **<65%**
Low	Tier 1	PLD 2				
	Tier 2	PLD 3				
Middle	Tier 3	PLD 4				
High	Tier 4	PLD 5				
	Tier 5	PLD 5*				

Approval of Student Growth Objective

Administrator approves scoring plan and assessment used to measure student learning.

Teacher ___________________ Signature______________________ Date Submitted________________

Evaluator __________________ Signature ______________________ Date Approved ________________

Results of Student Growth Objective

Preparedness Group		Students at Target Score	Teacher SGO Score	Weight (based on students per group)	Weighted Score	**Teacher SGO Score**
Low	Tier 1	PLD 2				
	Tier 2	PLD 3				
Middle	Tier 3	PLD 4				
High	Tier 4	PLD 5				
	Tier 5	PLD 5*				

Notes

Describe any changes made to SGO after initial approval, e.g. because of changes in student population, other unforeseen circumstances, etc.

Review SGO at Annual Conference

Describe successes and challenges, lessons learned from SGO about teaching and student learning, and steps to improve SGOs for next year.

Teacher ______________________ Signature ________________________ Date ____________________

Evaluator ______________________ Signature ________________________ Date ____________________

Scoring Details

Algebra8	Student ID	First Name	Last Name	Gender	MATH PARCC 2018	Conversion	EoC IED	Conversion	Getting to Know You Lab	Final Score	Initial Tier
					777	90	8	89	75	85	4
					765	90	2	22	85	66	2
					762	90	6	67	95	84	4
					772	90	3	33	65	63	2
					790	90	8	89	95	91	5
					771	90	6	67	70	76	3
					762	90	3	33	85	69	2
					768	90	5	56	65	70	3
					814	100	6	67	85	84	4
					763	90	4	44	75	70	3

Conversion Table

PARCC Score	Conversion	EOC Score	Conversion	Average	Tier Conversion
650 -699	60	1	11%	50 - 59	1
700 - 724	70	2	22%	60 - 69	2
725 - 749	80	3	33%	70 - 79	3
750 - 800	90	4	44%	80 - 89	4
800 - 850	100	5	56%	90 - 100	5
		6	67%		
		7	78%		
		8	89%		
		9	100%		