

Computer Science Performance Task

Performance Task Title: Creating a Computer Program

Content Area: Computer Science

Grade-Level: 7th Grade

Duration: 2-3 Weeks

Standards to be Assessed (3-5 standards):

Identify specific Standards of Learning ([SOLs](#)).

SOL Strand: Algorithms and Programming

7.1: The student will construct programs to accomplish a task as a means of creative expression or scientific exploration using block-based or text-based programming language, both independently and collaboratively,

a. combining control structures such as if-statements and loops including compound conditionals; and

b. creating clearly named variables that represent different data types, including numeric and non-numeric data, and perform operations on their values.

7.2: The student will document programs to make them easier to follow, test, and debug.

Learning Target/Assessment Goal:

Identify the observable or measurable outcomes desired.

- **I can** develop a flowchart using step-by-step algorithms of pseudocode for my computer program.
- **I can** define and apply loops in my computer program.
- **I can** define and apply conditional logic in my computer program.
- **I can** present my program and its functionality to an audience of experts and community members.

Assessment Task Description (Situation):

What is a meaningful context for students to consider in completing this task?

Students learn to use computational thinking (*decomposition, abstraction, pattern recognition, and algorithm design*) to solve computational problems in the ‘Computational Thinker’s Project’ unit plan. For this performance task, they will then apply computational thinking to create an original program and teach their coding skills to an authentic audience comprised of experts and community members. To complete the task, students will interact with CS experts, work in teams and assume the roles of either software engineers, software developers, or programmers.

Computer Science Performance Task

Acceptable Product(s) and/or Performance:

What will students produce as evidence of attainment of outcomes?

Students will create a computer program using a flowchart, loops, and conditional logic. They can choose to create their program from the following educational technology:

1. Sphero Bolt
2. LittleBits Code Kit
3. Scratch Jr
4. App Lab by Code.org

Authentic Audience:

The audience comprises an individual(s) interested in the findings and products that students create. They may score and or provide feedback about student performances based upon the products and presentations made.

We will invite computer science experts, community members, and other classes to view each student's performance task as a culminating event.

Student Role(s):

The Role provides the student with the opportunity to assume the career role or job associated with accomplishing the goal of the performance task.

Students may complete the performance task individually or in a team and may choose from the following three roles to complete their computer program:

1. Program Manager
2. Software Developer
3. Programmer

Student Task Instructions:

How will students complete this performance task? What activities will they do and to which standards and outcomes are they connected? What is the timeline? What materials or resources will they need?

Use the Backward Design Planning Tool below to organize formative assessments and lessons/scaffolds for each learning goal.

Computer Science Performance Task

Backward Design Planning Tool: See a completed example on [this link](#).

Final Product(s) Includes Presentations, Performance tasks, and Summative Assessments	Learning Goals (Targets/Objectives) Includes knowledge, understanding & skills needed by students to successfully complete products/tasks	Formative Assessments Formal and informal checks of understanding to ensure students are on track with the intended learning	Lessons, Instructional Strategies and Scaffolds These must be closely aligned to learning targets/objectives and formative assessments
Computer Program 2-3 Weeks	I can develop a flowchart using step-by-step algorithms of pseudocode for my computer program. (2-Class Periods)	1. Flowchart summary of algorithm logic for program 2. Program Rubric 3. Reflection in design journal 4. Exit ticket following daily lesson	1. Computer science guest speaker 2. Gliffy article on Flowchart universal symbols using the Connect, Extend, Challenge visible thinking routine.
	I can define and apply loops in my computer program. (2-Class Periods)	1. Loops quiz 2. Program Rubric 3. Reflection in design journal 4. Exit ticket following daily lesson	1. CT elements graphic organizer 2. Station Rotations using 'Workshop Model' structure <ul style="list-style-type: none"> a. Coding with teacher b. Coding with a peer (pair programming) c. Individual coding 3. Programming Video
	I can define and apply conditional logic in my computer program. (2-Class Periods)	1. Program rubric 2. Reflection in design journal 3. Exit ticket following daily lesson	1. Presentation rehearsal 2. Conditional logic video and simulation
	I can present my computer program and its functionality to an audience of experts and community members. (1-Class Period)	1. Presentation Rubric 2. Reflection following presentation	1. Presentation rehearsal 2. Program presentation

Computer Science Performance Task

Scoring (Program Rubric):

What is an appropriate tool for scoring this assessment?

Emerging 1-3 Provide feedback for improvement	Proficient 4 Grade Level Expectations Met Provide feedback for improvement	High Proficient 5 Provide feedback for improvement	SCORE
	I have developed a flowchart using step-by-step algorithms of pseudocode for my computer program. (7.2)		/5
	I have defined and applied loops in my computer program. (7.1 a)		/5
	I have defined and applied conditional logic in my computer program. (7.1 a)		/5
	I have presented my computer program and its functionality to an audience of experts and community members. (7.1 a, b, 7.2)		/5
Total			/20

Computer Science Performance Task

Teacher Reflection: *Following implementation, describe how the performance task can be improved.*