forward EDUCATION

Forward Education Lesson Rubrics

Lesson:	Student/Group:
Building/Coding Approach:	☐ **Use** ☐ **Modify** ☐ **Create**

Note: We recommend that you evaluate all students in the areas of "Real-World Application" and "Computer Science & Coding," as well as the approach (Use, Modify, Create) they used to complete their project.

This rubric is designed for flexible evaluations. Depending on how many phases students were asked to complete (e.g., Modify & Cr eate), leave the remaining areas blank (e.g. Use).

High School

4: Exemplary	3: Proficient	2: Developing	1: Novice	
Real-World Application				**Total**
Describes the real-world application of their project, using examples from the lesson and/or local contexts, with next steps to take action, including some career connections.	Describes the real-world application of their project, using examples from the lesson and/or local contexts and career connections.	Partially describes the connection to their project and its real-world application, with limited links to lesson concepts.	Struggles to describe the connection to their project and its real-world application.	
Computer Science & Coding				**Total**
Fully describes (verbally or using code comments) the process of intentionally testing and debugging their code using correct terminology. Includes clear evidence of how they optimized their program.	Describes (verbally or using code comments) the process of testing and debugging their code using some terminology, including some evidence of how they optimized their program.	Partially describes (verbally or using code comments) the process of testing and debugging their code using minimal terminology or minimal evidence of how they optimized their program.	Struggles to describe (verbally or using code comments) a process of testing or debugging their code. Does not provide evidence of how they optimized their program.	
Accurately uses visual tools (flowchart/algorithm) to describe their program and all of its features.	Uses visual tools (flowchart/algorithm) to describe their program and several of its features.	Struggles to use visual tools (flowchart/algorithm) to describe their program or one of its features.	Does not use a visual tool (flowchart/algorithm) to describe their program.	
Code functions without errors, includes at least two of the following: arrays, lists, complex functions, text-based programming, radio control, or multiple libraries.	Code functions with minimal errors and includes at least one of the following: arrays, lists, complex functions, text-based programming, radio control, or multiple libraries.	Code partially functions with several errors, and includes at least one of the following: arrays, lists, complex functions, text-based programming, radio control, or multiple libraries.	Code does not function, or does not include any of the following: arrays, lists, complex functions, text-based programming, radio control, or multiple libraries.	

4: Exemplary	3: Proficient	2: Developing	1: Novice	
Use - Understanding				**Total**
Confidently describes both the program and physical model using correct terminology, explaining how the sensors react to their environment.	Describes the general outcome of the code using some terminology. Identifies how at least one sensor responds to its environment.	Runs the code and builds the model, but requires significant prompting to explain how it works, or cannot identify how sensors respond to its environment.	Runs the code and builds the model, but cannot explain how it works, or how sensors respond to their environment.	
Modify - Tinkering				**Total**
Makes and tests changes to their program or physical model using an intentional approach, describing their rationale and program outcome using correct terminology.	Makes changes to their program or physical model following a tutorial, describing the outcome of the change with some details and terminology.	Modifies the project to perform a new task, but the code is redundant or contains unaddressed logical loops which they have not attempted to solve.	Modifications are limited to simple value changes (e.g. integers/strings) rather than structural logic changes.	
Create - Application				**Total**
Develops a fully functional, original system that utilizes multiple sensors and is clearly linked to several success criteria. Describes an intentional approach to their project build & code using correct terminology.	Develops a fully functional system using at least one sensor, linked to at least one success criterion. Describes some challenges or troubleshooting encountered, using some terminology.	Project is original in concept but lacks technical depth or relies on simple linear code rather than modular functions. The project is not clearly connected to the success criteria.	Project is unfinished or relies heavily on "copy-pasting" blocks from multiple unrelated tutorials that do not link to success criteria.	
Total:				