Lesson Planning Template **Teacher: DeSIRE**

📆Date:	Topic: Color Identify	Time Required: 90 minutes

🎯**Learning Target/Objectives:**

- **I can** program a robotic system to utilize a multi-sensor "handshake" (light sensor for detection followed by color sensor for identification) .
- **I can** explain the program logic used to branch robotic behavior based on detected color wavelengths .
- **I can** modify software parameters to reassign physical sorting destinations for specific identified objects .

📜**Vocabulary:**

- **Dual-Sensor Logic**
- **#define ACTIONGROUP**
- **Case 2 (Switch Statement)**
- **colorDetect() Function**
- **refreshNow**.
- **Sorting Destination**

❓**Guiding Questions:**

- Why does the robot use a light sensor to find an object before it uses the color sensor to identify it ?
- How does the `if-else if` logic chain prevent the robot from trying to sort a block into two places at once ?
- If the robot identifies a "Red" block but moves it to the "Blue" area, which line of code has been incorrectly programmed?
- Why is a 200ms timer added at the end of the sorting step before the robot returns to its search mode ?

📚**Lesson Design Details:**

- **Activity 1: The Sensor Handshake Trace**
  - **Focus:** Students manually move a block through the sorting cycle: 1. Pass the light sensor, 2. Wait at the color sensor, 3. Move to target. They must identify which line of code governs each step.
- **Activity 2: The "Middle Sort" Logic Challenge**
  - **Focus:** Students modify the code so that the Red block is placed in the "Middle" (ACTIONGROUP_GREEN) instead of the "Right" (ACTIONGROUP_RED). They must successfully upload and demonstrate the new behavior .
- **Activity 3: The OLED Display Audit**
  - **Focus:** Students change the OLED display logic. They must modify the program so that when a Blue block is detected, the screen displays their own name instead of "BLUE" (keeping the 6-character limit in mind).

Key Points (Vocabulary):

- **Dual-Sensor Logic:** A programming method where two different sensors must provide data in sequence to complete a task .
- **#define ACTIONGROUP:** A command used to assign a functional name (like RED, GREEN, BLUE) to a specific numerical action file in the robot's memory.
- **Case 2 (Switch Statement):** A section of code that manages specific "scenarios," such as what the robot does when it identifies an object.
- **colorDetect() Function:** The specific sub-routine in the code that tells the color sensor to read the wavelength of the block.
- **refreshNow:** A boolean (True/False) variable used to tell the OLED screen to update its display result .
- **Sorting Destination:** The physical grid area on the map where the robot places an object after identification .

Key Points of Instruction

- **Logical Sequencing:** Emphasize that the light sensor is the "Gatekeeper." If the light sensor doesn't detect an object first, the rest of the sorting code will never run .
- **Variable Mapping:** Help students understand that names like `ACTIONGROUP_RED` are linked to specific numbers (like 22). If they change the number, they change the robot's physical destination.
- **Serial Port Discipline:** As with previous lessons, selecting the correct COM port and board (Arduino Uno) is mandatory for a successful "Done uploading" prompt .
- **Hardware/Software Balance:** Remind students that mechanical placement on the map is just as important as the code; the blocks must be placed exactly where the sensors can see them.

Teacher's Cheat Sheet	
Parameter	**Technical Requirement / Data Point**
Sketch File	color_sorting.ino
IDE Space	Approx. 24,406 bytes (75% of storage)
Light Sensor Role	Detect presence of any object
Color Sensor Role	Identify color of the detected object

Red Target (Default)	Action Group 22 -> Right Area
Green Target (Default)	Action Group 23 -> Middle Area
Blue Target (Default)	Action Group 24 -> Left Area
Upload Alert	Do not move USB cable during "burning"

Category	Standard Organization	Standard/Benchmark Code and Description
Technology	ITEEA	**STEL-2R:** Follow step-by-step instructions to safely use systems and troubleshoot common problems
Computer Science	NCSOS	**HS-CS-03:** Illustrate the ways computing systems implement logic, input, and output through hardware components
Computer Science	NCSOS	**HS-AP-14:** Create procedures with parameters to organize code and make it easier to reuse
Digital Literacy	ISTE	**1.5.d:** Students understand how automation works and use algorithmic thinking to develop a sequence of steps
Engineering	ITEEA	**STEL-2V:** Analyze the stability of a technological system and how it is influenced by components in the feedback loop