**Lesson Planning Template** **Teacher: DeSIRE**

| 📆Date: | Topic: Auto Sorting | Time Required: 90 minutes |
|---|---|---|

🎯**Learning Target/Objectives:**

- **I can** successfully configure the Arduino IDE to compile and upload autonomous sorting logic to a robotic controller .
- **I can** explain how an IR obstacle avoidance sensor uses infrared light to detect the presence of an object without physical contact.
- **I can** execute a fully automated sorting sequence where a robot independently recognizes, grips, and transports objects to specific grid locations .

📜**Vocabulary:**

- **IR Obstacle Avoidance**
- **Autonomous Sorting**
- **Non-Contact Detection:**
- **Ino Sketch**
- **Debugging Prompt**
- **Map Placement**

❓**Guiding Questions:**

- How does an infrared sensor "see" a color block differently than a human eye does?
- Why is "Auto Sorting" more efficient for a factory than a "Claw Machine" style manual trigger?
- What environmental factors (like bright sunlight) might interfere with the accuracy of an IR obstacle avoidance sensor?

📚**Lesson Design Details:**

- **Activity 1: The IR Visibility Lab**
  - **Focus:** Students use the Serial Monitor to observe the IR sensor's "raw data." They test different materials (clear plastic, black paper, shiny metal) to see which blocks reflect IR light best and trigger the sorting logic.
- **Activity 2: Autonomous Grid Challenge**
  - **Focus:** Students place blocks in three different starting positions on the map. They must verify that the robot correctly "recognizes" each block and transports it to the intended target location without any button presses .
- **Activity 3: The "Wait then Move" Logic Edit**
  - **Focus:** Students navigate the code to find where the transport action is triggered. They attempt to add a "delay" so the robot waits 3 seconds after detecting a block before it begins to move.

**Key Points (Vocabulary):**

- **IR Obstacle Avoidance:** A sensor that emits infrared light and detects the reflection to "see" if an object is ahead.
- **Autonomous Sorting:** The process where a machine identifies and organizes items without human intervention.
- **Non-Contact Detection:** Identifying an object's presence without physically touching it.
- **Ino Sketch:** The source code file (`auto_sorting.ino`) that contains the robot's sorting instructions.
- **Debugging Prompt:** The area at the bottom of the software that provides feedback on the success of a code upload.
- **Map Placement:** The specific physical alignment of the robot and sensor kit required for the transport logic to function.

## Key Points of Instruction

- **Infrared Principles:** Briefly explain that the sensor sends out invisible IR light. When a block is placed in front, that light bounces back to a receiver, signaling the robot to start the "Grab" sequence .
- **Code as a Trigger:** Teach students that the `auto_sorting.ino` program is essentially "listening" for a signal from the IR sensor. Once it gets that signal, it stops listening and starts moving.
- **Port Management:** Remind students that the "COM port" is the specific "doorway" the computer uses to talk to the Arduino board. If they select the wrong port, the robot will not receive the sorting instructions .
- **Mechanical Clearance:** Before clicking "Run," students must ensure the robotic arm has a clear path to all grid positions on the map to prevent hardware crashes .
- **Pro-Tip for Educators:** If the robot won't stop moving even when no block is there, the IR sensor might be detecting the "floor" or map itself. Adjust the physical angle of the sensor or the sensitivity knob if applicable.

| **Teacher's Cheat Sheet** | |
|---|---|
| **Feature** | **Technical Requirement / Data** |
| **Sketch Path** | auto_sorting -> auto_sorting.ino |
| **Hardware Brain** | Arduino / Genuino Uno |

| | |
|---|---|
| **Memory usage** | Uses approx. 14,262 bytes (44% of storage) |
| **Trigger Type** | IR reflection (Obstacle Detection) |
| **Upload Status** | Wait for "Done uploading" in the prompt area |

| 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 |
| **Engineering** | ITEEA | **STEL-2V:** Analyze the stability of a technological system and how it is influenced by components in the feedback loop |
| **Digital Literacy** | ISTE | **1.5.d:** Students understand how automation works and use algorithmic thinking to develop a sequence of steps |
| **Science** | NCSOS | **PSc.3.2.1:** Explain the properties of electromagnetic waves (Infrared) and their use in technology. |