Lesson Planning Template **Teacher: DeSIRE**

📅**Date:**	**Topic: Claw Machine**	**Time Required: 90 minutes**

🎯**Learning Target/Objectives:**

- **I can** configure the Arduino development environment to compile and upload specialized robotic "Claw Machine" logic .
- **I can** explain how a touch sensor acts as an "interrupt" or trigger for a robot engaged in continuous motion .
- **I can** execute a multi-phase robotic operation that includes searching (rotating), detecting (sensor touch), and transporting (specified positioning) .

📜**Vocabulary:**

- **Claw Machine Logic**
- **Continuous Rotation**
- **Event-Driven Programming**
- **Ino Sketch**
- **Target Area**
- **Compilation**

❓**Guiding Questions:**

- How does the robot "know" to keep moving left and right until the sensor is touched?
- Why is timing more critical in a "Claw Machine" simulation than in the previous "Touch Control" sorting lesson?
- What would happen to the robot's mechanical stability if the "transport" speed was set too high in the code?

📚**Lesson Design Details:**

- **Activity 1: The Code Trace**
  - **Focus:** Students open the Doll_machine.ino sketch and search for the loop that controls the "left and right" rotation. They must describe in plain English what triggers the change from "rotating" to "gripping."
- **Activity 2: The Precision Operator Challenge**
  - **Focus:** Students set up a "Doll Machine" station with multiple small objects. They must successfully clear the board by timing their sensor touches as the arm sweeps over the targets.
- **Activity 3: Troubleshooting "The Frozen Arm"**
  - **Focus:** The teacher provides a setup where the sensor is unplugged or the port is wrong. Students must use the Arduino IDE debugging area to identify why the arm isn't rotating or responding to touch.

Key Points (Vocabulary):

- **Claw Machine Logic:** A specific programming pattern where a robot moves back and forth until a user signals it to drop and grab.
- **Continuous Rotation:** A state where the robot arm segments move repeatedly between two set angular points.
- **Event-Driven Programming:** Code that remains in a loop until a specific external stimulus (the sensor touch) occurs.
- **Ino Sketch:** The primary code file used in the Arduino IDE to define robot behaviors.
- **Target Area:** The specific physical location on the map where an object must be gripped or placed.
- **Compilation:** The process where the IDE checks the code for syntax errors before sending it to the robot .

Key Points of Instruction

- **Synchronization of Motion:** Point out that the robot is programmed to perform two distinct behaviors: a loop for searching (rotating) and a single action for the "grip" .
- **The Tools Menu "Check":** Students frequently experience upload errors because they forget to re-verify the "Port." Ensure they check that their COM port is still active before every upload .
- **Physical Mapping:** Before running the code, students must physically align the map so the robot's "left and right" sweep actually passes over the intended target blocks.
- **Software Confirmation:** Teach students to recognize "Done compiling" as a signal of logical accuracy and "Done uploading" as a signal of hardware success.

Teacher's Cheat Sheet

Parameter	Technical Requirement
Code Directory	Doll_machine -> Doll_machine.ino23.
Board Selection	Arduino / Genuino Uno24.
Port Verification	Tools -> Port (Select active COM port)25252525.
Compilation	Checkmark icon (wait for "Done compiling")26.

Upload	Arrow icon (wait for "Done uploading") 27.
Hardware Link	Keep USB cable still during the "verify" and "upload" phase28.
Safety Warning	Do not move the USB cable during upload
Serial Connection	5V via USB cable

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-13:** Decompose problems into smaller components through systematic analysis.
Digital Literacy	ISTE	**1.5.d:** Students understand how automation works and use algorithmic thinking to develop a sequence of steps
Engineering	ITEEA	**STEL-7Q:** Apply a broad range of making skills to follow a design process in the construction of a prototype