Lesson Planning Template **Teacher: DeSIRE**

📆Date:	Topic: Mobile APP Programming	Time Required: 90 minutes

🎯**Learning Target/Objectives:**

- **I can** use the "Motor Off" function to manually manipulate a robotic arm into specific postures for programming.
- **I can** record a sequence of poses to create a complete automated action group using a mobile application .
- **I can** debug and modify individual action parameters, such as delay time and posture angles, to optimize robotic performance .

📜**Vocabulary:**

- **Kinesthetic Programming (Manual Teaching**
- **Motor Off (Power Down)**
- **Posture Angle**
- **Action Group**
- **Loop**
- **Milliseconds (ms)**

❓**Guiding Questions:**

- Why is it dangerous to perform a "Motor Off" operation if the robotic arm is not on a smooth, open surface?
- How does the "Add Action" button act like a digital camera for the robot's physical position?
- If your robot moves too fast and tips over, which specific parameter in the "Create Actions" list should you modify?

📚**Lesson Design Details:**

- **Activity 1:** The "Simon Says" Calibration
  - **Focus:** Students use the "Motor Off" mode to move the robot into five classic poses (Wave, Bow, Reach, Grip, Release). They must record each pose and then use "Run Action Group" to verify the sequence .
- **Activity 2:** The Millisecond Slow-Down
  - **Focus:** Students program a "fast" action at 200ms and a "slow" action at 2000ms. They must observe the difference in stability and explain why certain tasks require higher time values.
- **Activity 3:** The Collaborative Action Group
  - **Focus:** Student A creates the first 3 actions of a sequence and saves it. Student B opens the file in "My Action," adds 2 more actions, and optimizes the loop .

Key Points (Vocabulary):

- **Kinesthetic Programming (Manual Teaching):** A method of programming where the user physically moves the robot's joints into position while the motors are disengaged.
- **Motor Off (Power Down):** A command that releases the holding torque of the servos, allowing the arm joints to become loose for manual adjustment.
- **Posture Angle:** The specific numerical value assigned to each of the six servos that determines the arm's physical stance.
- **Action Group:** A saved collection of multiple robotic poses played back in a specific sequence.
- **Loop:** A programming command that tells the robot to repeat an action group continuously.
- **Milliseconds (ms):** A unit of time used to set the delay between actions; 1000ms equals 1 second.

Key Points of Instruction

- **Permission Management (Android):** Emphasize that Android users must enable all app permissions in their phone settings before starting. If they don't, their hard work might not be saved.
- **The "Limp Arm" Safety Warning:** When students click "Motor Off," the robot loses all internal strength. Instruction must focus on holding the arm steady so it doesn't crash into the base or the table.
- **Recording Sequence:** Students must remember the "Move then Add" rhythm. They physically move the arm, then click "Add Action" to record that specific data point.
- **Time Conversion:** Ensure students understand that the robot's "speed" is controlled by time. A higher millisecond value results in a slower, more controlled movement.

Teacher's Cheat Sheet

Parameter	Data Point / Requirement
App Name	Wonderbot
Android Prep	Enable all permissions in system settings first
Safety Step	Ensure arm is in an open area before clicking "Motor Off"
Programming Rhythm	Disengage Motor -> Move Arm -> Click Add Action

Time Conversion	1000ms equals 1 second
Default Time	Usually 1000ms for steady movement
Debugging	Click specific row on the left to modify or delete that step

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-AP-10: Create procedures with parameters to organize code and make it easier to reuse
Engineering	ITEEA	STEL-7Q: Apply a broad range of making skills to follow a design process in the construction of a prototype
Mathematics	NCSOS	NC.M1.G-CO.2: Represent transformations in the plane; identifying specific coordinate-like values for servo positions
Digital Literacy	ISTE	1.5.d: Students understand how automation works and use algorithmic thinking to develop a sequence of steps