Lesson Planning Template **Teacher: DeSIRE**

📆Date:	Topic: Integrate Action Files	Time Required: 120 minutes

🎯**Learning Target/Objectives:**

- **I can** use the integrate function to merge multiple action group files into a single, continuous series of movements.
- **I can** identify and manage hardware constraints when a programmed sequence exceeds the controller memory limit.
- **I can** troubleshoot and debug complex action lists by running them online before final file saving.

📜**Vocabulary:**

- **Integrate**
- **Series Connection**
- **Action Details List**
- **Run Online**
- **Memory Limit**
- **Modular Programming**

❓**Guiding Questions:**

- Why is it more efficient to program small "modules" of movement and integrate them later rather than programming one giant file from scratch?
- If the robot performs well "Online" but cannot be "Downloaded," what technical constraint have you likely hit?
- How does the order in which you select files for integration change the final "story" of the robot's movement?

📚**Lesson Design Details:**

- **Activity 1:** The "Robot Storyteller" (Modular Design)
  - **Focus:** Students are given three separate files (e.g., Wave, Bow, Grip). They must use the Integrate function to create a "Robot Greeting" sequence. They then experiment with changing the order of integration to see how it changes the robot's "personality".
- **Activity 2:** Constraint Management Challenge
  - **Focus:** Students deliberately merge so many files that they exceed the 1020-action limit. They must observe the software prompt and then practice "editing down" the list to make it small enough to download.
- **Activity 3:** Integration Debuggers
  - **Focus:** Pairs of students trade integrated files that have a "glitch" (a jerky movement between merged files). They must use "Run Online" to find the error in the Action Details List and fix it.

Key Points (Vocabulary):

- **Integrate:** The software function that allows a user to merge two or more separate action files into one long series.
- **Series Connection:** The state of having multiple movement sequences linked together in a specific order.
- **Action Details List:** The central table in the software that displays every specific servo position and time duration in a sequence.
- **Run Online:** A mode where the robot performs the sequence while connected to the PC, allowing for real-time testing.
- **Memory Limit:** The maximum number of individual actions (1020) that the controller can physically store.
- **Modular Programming:** Breaking a large task into smaller, reusable pieces (like No. 1 and No. 2 action groups).

Key Points of Instruction

- **Sequential Merging:** Explain that the first file opened becomes the base of the list, and every subsequent file added via the "Integrate" button is tacked onto the end.
- **The 1020 Threshold:** This is the most critical technical constraint. Students must be aware that if their integrated list exceeds 1020 actions, they can only run the robot while it is tethered to the PC; they cannot save it for standalone use.
- **Verification Before Saving:** Teach students to always "Run Online" after an integration to ensure the transition between the two merged files is smooth and does not cause a mechanical collision.
- **File Management:** Emphasize the importance of clear naming conventions, such as "121 Rotation" or "No. 121 Grip," to distinguish integrated files from original single-action files.

Teacher's Cheat Sheet	
Technical Parameter	**Data Point / Constraint**
Integration Button	Located at the bottom of the "Run online" section
Max Downloadable Actions	1020 individual actions
Software Prompt	Triggers when the list is greater than 1000/1020 actions
Action File Format	.rob file extension

Online Testing	Necessary to check transitions before final save
Naming Integrated Files	Must be distinct to avoid overwriting original modules

Category	Standard Organization	Standard/Benchmark Code and Description
Computer Science	NCSOS	HS-AP-10: Create procedures with parameters to organize code and make it easier to reuse11.
Technology	ITEEA	STEL-2R: Follow step-by-step instructions to safely use systems and troubleshoot common problems2222.
Engineering	ITEEA	STEL-3H: Optimize a system by identifying and managing various constraints3.
Digital Literacy	ISTE	1.5.d: Students understand how automation works and use algorithmic thinking to develop a sequence of steps4444.
Mathematics	NCSOS	NC.M1.G-CO.2: Represent transformations in the plane through coordinate-based servo values5.