Unit 2: Python with JuiceMind: Smart Doorbell & Ping Pong Launcher

Content Area:	**Computer Science**
Course(s):	
Time Period:	**Marking Period 1**
Length:	**15-20 days**
Status:	**Published**

Summary

This unit has students apply the Python fundamentals introduced in Unit 1 — data types, variables, operators, and conditional (if/else) logic — to build two real, physical devices using JuiceMind's guided online coding kits. Students first build a Smart Doorbell: a circuit built from a microcontroller, breadboard, buttons, and LEDs that sends a text message or email when its button is pressed. Students then build a Ping Pong Launcher, which adds a servo motor (to set a firing angle between 0 and 90 degrees) and a DC motor (to fire a ping pong ball up to 20 feet), extending the same programming concepts into basic robotics and motor control. Both projects require students to design, wire, code, test, and troubleshoot a combined hardware-and-software system, giving them direct, hands-on experience with the engineering design process alongside their growing Python skills.

Revision Date: July 2026

Essential Questions

- How do hardware and software work together to create a functioning device?
- How do conditional statements (if/else) let a program respond differently to different real-world inputs, such as a button press?
- What does it mean to design, test, and troubleshoot a system, and how is troubleshooting hardware different from troubleshooting code alone?
- How do engineers balance function, cost, and constraints when designing a physical product?

Enduring Understandings

- A functioning device depends on the coordinated design of both its physical components (hardware) and the instructions that control them (software).
- Conditional logic allows a program to sense and respond to the physical world, turning a simple input

(like a button press) into a meaningful output (like a notification or a motor firing).

- Testing, troubleshooting, and iterating are essential, expected parts of building any real system; effective troubleshooting isolates whether the hardware, the software, or both are the source of unexpected behavior.

Objectives

Students Will Know

- Variables and data types are used to store and represent different kinds of information in a program.
- Comparison and logical operators, and how they are used inside conditional (if/else) statements to control a program's behavior.
- How a microcontroller, breadboard, buttons, LEDs, and other electronic components connect to form a working circuit.
- How a program communicates with hardware inputs (buttons/sensors) and outputs (LEDs, motors, notifications).
- The difference between a servo motor (fixed-position control, e.g., setting the launcher's angle) and a DC motor (continuous rotation, e.g., firing the ball).
- The engineering design process: define the problem, design, build, test, troubleshoot, and iterate.

Students Will be Skilled At

- Writing Python code using variables, data types, and operators to store and manipulate real values.
- Writing and testing if/else conditional statements that change a program's behavior based on an input.
- Defining and calling functions to organize code that controls a physical device.
- Assembling a circuit on a breadboard according to a wiring diagram.
- Connecting code to hardware so that a physical input (a button press) produces a physical or digital output (an LED, a notification, a motor movement).
- Systematically troubleshooting a combined hardware/software system when it does not behave as expected.
- Documenting and explaining design decisions and trade-offs made while building a physical product.

Learning Plan

Because JuiceMind is a project-based, self-paced online platform rather than a sequence of individually numbered lesson plans like CodeHS, this unit is organized around two hands-on builds rather than a daily lesson-by-lesson breakdown. Both projects follow the same general arc: guided online tutorial and concept instruction, circuit assembly, incremental coding milestones, integration and testing, and a short design reflection. The teacher should preview the live JuiceMind tutorials before teaching each project to confirm current pacing, built-in checkpoints, and any updated safety guidance for the motorized launcher.

Project 1: Smart Doorbell (estimate 5-8 days)

Students work through JuiceMind's guided online tutorial to build a Smart Doorbell using a microcontroller, breadboard, two buttons, two LEDs, and connecting wires. As they progress, students apply core Python concepts from Unit 1 — data types, variables, and operators — and use if/else conditional statements to program the doorbell's core logic: when the button is pressed, the program checks the input and triggers an output, first an LED, and ultimately a text message or email notification. Along the way, students assemble and test their circuit, debug wiring and code errors, and use functions to organize their program. The project concludes with students explaining how their circuit and code work together and briefly reflecting on a design trade-off or troubleshooting challenge they encountered.

Project 2: Ping Pong Launcher (estimate 8–10 days)

Building on the doorbell project, students construct a Ping Pong Launcher using the same microcontroller and breadboard alongside a servo motor, which sets the launcher's firing angle between 0 and 90 degrees, and a DC motor, which fires the ball. Students extend their Python skills — variables, operators, conditionals, and functions — to program button-controlled inputs that adjust the servo's angle and trigger the DC motor to launch a ping pong ball up to 20 feet. As with the doorbell, students build the physical mechanism from the kit, wire it according to a diagram, write and test the controlling code, and troubleshoot both mechanical and code-based issues as they arise. The project closes with a live test/demonstration and a short reflection in which students explain their build, describe a problem they had to troubleshoot, and identify a design trade-off (e.g., firing angle vs. distance, power vs. accuracy).

Assessment

Formative Assessment

In-platform tutorial checkpoints and auto-checked coding steps within JuiceMind's guided IDE, teacher circulation and spot-checks during circuit assembly, informal troubleshooting conversations, and beginning/end-of-class discussion questions.

Summative/Performance Assessment

A completed and functioning Smart Doorbell that sends a notification when its button is pressed, and a completed and functioning Ping Pong Launcher that fires a ball to a target angle/distance, each evaluated against a rubric addressing code correctness (variables, conditionals, functions), circuit assembly, and overall functionality.

Alternative Assessment

A short written or verbal design reflection for each project in which students explain what their device does, describe one troubleshooting challenge, and identify one design trade-off they navigated.

District-Level/Benchmark Assessment

Unit 2 Benchmark Assessment, administered at the close of the unit after both builds are complete. Part A is a 10–12 question multiple-choice/short-answer section assessing conceptual understanding of variables and data types, comparison/logical operators, if/else conditional logic, how code communicates with hardware inputs and outputs, the difference between servo and DC motor control, and the steps of the engineering design process (define, design, build, test, troubleshoot, iterate). Part B is a short applied troubleshooting task in which students are given a doorbell or launcher circuit and code containing one deliberately introduced error (a miswired component or a logic bug) and must identify and correct it within a set time, scored against a rubric for accurate diagnosis, correct fix, and clear explanation of the reasoning used.

Materials

Core instructional materials

- JuiceMind online platform account/license, including the guided Smart Doorbell and Ping Pong Launcher tutorials
- JuiceMind Smart Doorbell Coding Kit (microcontroller, breadboard, buttons, LEDs, wires) — one per student or student pair
- JuiceMind Ping Pong Launcher Coding Kit (microcontroller, breadboard, servo motor, DC motor, buttons, LEDs, wires, launcher hardware) — one per student or student pair

Supplemental materials

- USB cables and power source for each kit
- Safety guidelines for handling motorized/launching components
- Student laptops/Chromebooks with internet access
- Target markers or measuring tape for launcher distance testing
- Teacher-created design-reflection handout/template

Standards

8.1 Computer Science (2020 NJSLS Computer Science & Design Thinking, Grades 6–8)

8.1.8.AP.2 – Create clearly named variables that represent different data types and perform operations on their values.

8.1.8.AP.3 – Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

8.1.8.AP.9 – Document programs in order to make them easier to follow, test, and debug.

8.1.8.CS.1 – Recommend improvements to computing devices in order to improve the ways users interact with the devices.

8.1.8.CS.2 – Design a system that combines hardware and software components to process data.

8.1.8.CS.3 – Justify design decisions and explain potential system trade-offs.

8.1.8.CS.4 – Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.

8.1.8.NI.1 – Model how information is broken down into smaller pieces, transmitted as addressed packets through multiple devices over networks and the Internet, and reassembled at the destination (applied to how the Smart Doorbell sends a text/email notification).

8.2 Design Thinking (2020 NJSLS Computer Science & Design Thinking, Grades 6–8)

8.2.8.ED.1 – Evaluate the function, value, and aesthetics of a technological product or system, from the perspective of the user and the producer.

8.2.8.ED.3 – Develop a proposal for a solution to a real-world problem that includes a model (e.g., physical prototype, graphical/technical sketch).

8.2.8.ED.4 – Investigate a malfunctioning system, identify its impact, and explain the step-by-step process used to troubleshoot, evaluate, and test options to repair the product in a collaborative team.

8.2.8.ED.6 – Analyze how trade-offs can impact the design of a product.

8.2.8.ED.7 – Design a product to address a real-world problem and document the iterative design process, including decisions made as a result of specific constraints and trade-offs (e.g., annotated sketches).

9.4 Career Readiness, Life Literacies & Key Skills (2020 NJSLS)

9.4.8.CT.2 – Develop multiple solutions to a problem and evaluate short- and long-term effects to determine the most plausible option (applied to design trade-offs in both builds).

Interdisciplinary: English Language Arts (NJSLS-ELA, Grade 8)

SL.8.4 – Present claims and findings, sequencing ideas logically (applied to explaining a finished device and its design process).

Integrated Accommodations & Modifications

Suggested Strategies for Modification for Computer Science