



Mission 4: Flight Safety		Time Frame: 60-90 minutes	
<p>Mission Goal: Students will program a set of safety procedures that can be used in future drone flights.</p> <p>Learning Targets</p> <ul style="list-style-type: none"> • I can write code to arm the drone for take-off. • I can write code for a warning indicator. • I can understand the quadcopter power system. 		<p>Key Concepts</p> <ul style="list-style-type: none"> • Safety procedures can ensure the user is safe before a drone take-off. • CodeAIR has two input buttons. • An <i>if</i> and <i>elif</i> statements allow a branch of indented code to run, based on a condition. • The <i>break</i> statement stops a loop. • Buttons can accidentally record two presses. Code to “debounce” a button is necessary. • Create a function for code you want to reuse. 	
<p>Assessment Opportunities</p> <ul style="list-style-type: none"> • Quiz after Objective 2 • Quiz after Objective 5 • Complete the program <i>safety.py</i> • Mission 4 Assignment • Mission 4 Review questions 		<p>Success Criteria</p> <ul style="list-style-type: none"> <input type="checkbox"/> Use an <i>if</i> statement and <i>break</i> statement to stop a loop when a button is pressed <input type="checkbox"/> Use code to “debounce” button presses <input type="checkbox"/> Create a user-defined function <input type="checkbox"/> <i>safety.py</i> works correctly and runs without errors <input type="checkbox"/> Complete Mission 4 Assignment 	
Standards			
CSTA Standards Grades 9-10		CSTA Standards Grades 11-12	
<ul style="list-style-type: none"> • 3A-CS-03 • 3A-DA-11 • 3A-AP-13 • 3A-AP-16 • 3A-AP-17 • 3A-AP-21 		<ul style="list-style-type: none"> • 3B-NI-04 • 3B-AP-08 • 3B-AP-14 • 3B-AP-15 • 3B-AP-22 • 3B-AP-23 	
AI4K12 Standards Grades 9-12			
Student Materials		Teacher Resources	
<ul style="list-style-type: none"> • Laptop/computer with Chrome browser • CodeAIR drone and USB cable • CodeAIR Mission 4 Assignment • CodeAIR Flying Guide 		<ul style="list-style-type: none"> • CodeAIR Mission 4 Assignment Answers • CodeAIR Mission 4 Review Questions • CodeAIR Flying Guide 	
Vocabulary			
Quadcopter Safety Guidelines	Steps to take to ensure personal safety when working with a drone. They include wearing protective gear, avoiding contact with moving parts, and operating in a clear area.		
Safety interlock	A safety measure that prevents an electronic device from starting until an event is triggered, like a button press.		
UX	User experience; it encompasses the navigation of a product and how easy to use it is.		
Branching 'if' statement	A programming control structure that lets code do something different if a certain condition happens, like a button press. This is different from sequential or iteration.		
Bounce	When the metal contacts of an electronic input peripheral like a button bounce a few times before coming to a rest. This problem could mean a peripheral is read more than once.		



Function	Reusable code with a name. Making reusable components is a major goal of software engineering. Once a function is defined, it must be called before the code is executed.
Variable	A named value used in code, like a box with a label. Use the variable name instead of the value. A value can be any data type, including a number, a string (text) or a Boolean.
Torque	Rotational force produced by motors. When torque is produced, there is a naturally occurring force in the opposite direction.

New Python Code

<code>buttons.was_pressed(BTN_0)</code>	Checks to see if B0 was pressed since the last check.
<code>break</code>	Breaks out of the nearest enclosing loop
<code>if buttons.was_pressed(BTN_0): break</code>	If statement (branching) that checks for a button press. <code>buttons.was_pressed(BTN_0)</code> is either True or False.
<code>while True: if buttons.was_pressed(BTN_0): break</code>	If statement in an infinite loop. The code waits for a button press before moving to the next line of code.
<code>pixels.fill(YELLOW)</code>	Sets all 8 pixels to YELLOW (built-in color)
<code>pixels.off()</code>	Turn off all 8 pixels
<code>sleep(0.1) buttons.was_pressed()</code>	Debounce the buttons. This line of code resets both buttons!
<code>from flight import *</code>	Imports the flight module so you can use built-in functions, like <code>motor_test()</code>
<code>motor_test(True) motor_test(False)</code>	Start / stop a motor test that spins the motors but not fast enough to lift off.
<code>def button_arm():</code>	Function definition. The indented block below is the code of the function. A function definition always has <code>()</code> for parameters, even if none are given.
<code>return do_launch</code>	Returns (sends) data from the function back to the code that called it. A return ends the function.
<code>if button_arm():</code>	Call the function <code>button_arm()</code> , which returns a True or False value
<code>set_param('motorPowerSet.m2', 30000)</code>	Set motor (m2) power (30000)
<code>set_param('motorPowerSet.enable', 1)</code>	Enable power to the motors
<code>set_param('motorPowerSet.enable', 0)</code>	Disable power to the motors

Teacher Notes

- The hints in Objective 1 give suggestions for troubleshooting programming errors, and also for using Chat GPT to understand code. You may want to review the hints with students.
- The assignment document follows the mission and is a place for guided notes. You can print the document for the students (PDF) or assign it digitally through your LMS (doc).



- If you or your students have trouble running code, follow the CodeAIR Flying Guide. The CodeAIR must be connected and turned on, even if you are not flying the drone.
- The buttons to press are very tiny and can be hard to see. They are located just next to the first and last blue LEDs. The blinking lights help identify which button to press in the correct order.
- Review questions can be used as a class review, made into a Kahoot!, or used to create an exam in your learning management system.
- Extensions and cross-curricular projects are included to enhance the concepts in the mission. You can use the extensions to extend students' programming experience. A remix is planned after Mission 5.

Extensions

- Have students use ChatGPT to annotate a section of their code. Then have students do their own annotation of a different section of code.
- Have students do a code review. They should go through the `button_arm()` function and describe what each section does.

Cross-Curricular

- **SCIENCE:** The mission discusses torque and Newton's 3rd Law. Study these topics in more depth.
- **LANGUAGE ARTS:** After using Chat GPT to annotate a section of code, write a paragraph that describes how artificial intelligence drives many software and physical systems.
- **LANGUAGE ARTS:** Make a list of troubleshooting strategies used to identify and fix errors.