The Lab
Junior Program

Term 2
Home Workbook

Locations:

Kinex Mall
11 Tanjong Katong Road
#03-01/02 Singapore 437157

Taste Orchard
160 Orchard Road #03-07
Singapore 238842

Woods Square
12 Woodlands Square #04-68 Tower 1 Singapore 737715

Buangkok Square
Blk 991 #03-07
Singapore 530991

Preface

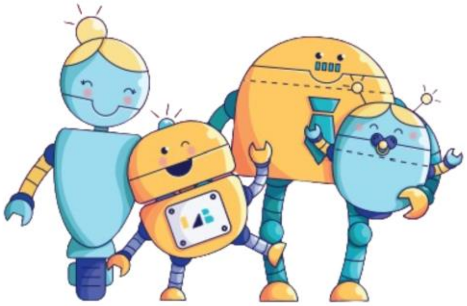
Version Oct 2025

Welcome to The Lab Singapore's workbook—a guide designed to spark creativity, nurture logical thinking, and develop essential coding skills.

At The Lab, we believe that every child can achieve remarkable things when given the right tools and guidance. Our passionate educators, driven by a love for teaching and innovation, are dedicated to empowering students to solve real-world problems through coding and computational thinking.

This workbook reflects our commitment to making learning engaging, meaningful, and fun. It is more than a collection of exercises—it's a step towards building confidence, resilience, and a lifelong love for learning.

Let's embark on this exciting journey together!



A Friendly Note to Parents & Students

This workbook is designed to help our students revise and practise the concepts taught in class. Many of these ideas are new, so it's completely normal for students to need repetition and extra practice before they fully understand them. Not getting it the first time is part of the learning journey — and absolutely okay!

Please remember that this workbook is not a test and it does not reflect how well your child is performing in class or how well we teach. It simply gives students a chance to try things on their own at home, at their own pace.

To keep things encouraging, here's how we look at the scores:

✓ Scoring (Just for Practice!)

50% and above – Great job!
Your child has understood the main ideas.

Below 50% – Keep practising!
This is not a failure. It just shows which areas are still new or need a bit more time — and we'll continue to guide them in class.

At The Lab, we believe learning should feel positive, safe, and joyful. We hope this workbook supports your child as they grow, explore, and discover new concepts!

Answer Key

Relational Operators with Conditionals

Coding Concepts

1. Answer: distance, cm. The ultrasonic sensor measures distance and the unit shown is cm(centimeters).

2. Answer: :Beep , 0.2. The block is a play beep block. It plays a beep from a keyboard for 0.2 seconds.

3. Answer: Both , 50%. The block controls the robot's movement motors, which means it starts both motors of the robots moving at a speed of 50%.

4. Answer: Stops. The block shown is stop moving, so the missing word is stops.

5. Answer: Greater/More . 100. The block shown is the comparison block (>) in LEGO SPIKE. It checks if the first value is greater than or more than 100, returning True or False.

6. Answer: B. The ultrasonic sensor sends sound waves and measures how long they take to bounce back. It is used to measure distance to object.

7. Answer: B. The ultrasonic sensor measures distance. The robot can stop or turn when it senses something nearby.

8. Answer: A. In LEGO SPIKE, the ultrasonic sensor displays distance in cm.

9. Answer: C. Only the ultrasonic sensor can sense distance, so it can detect how close the robot is to obstacles.

10. Answer: B. LEGO SPIKE Ultrasonic Sensor can measure distances starting from 4 cm up to about 200 cm. At 0 cm, it cannot detect because the sound cannot bounce back properly.

11. Answer: C. The LEGO SPIKE Ultrasonic Sensor can measure from 4 cm up to 200 cm. It cannot measure infinity, and 15 cm or 50 cm are too short for its maximum range.

12. Answer: Right. The right code makes a fast-beeping sound because the wait time is only 0.2 seconds, so the beeps repeat quickly. The left code waits 1 second, making the beeps slow.

13. Answer: Left. The left code makes a slow beeping sound because it waits 2 seconds between each beep, so the beeps happen slowly. The right code only waits 0.2 seconds, making the beeps faster.

14. Answer: Left. The left code matches the original because it uses "distance in cm < 15", which means "closer than 15 cm." The right code uses "distance in cm > 15", which means "farther than 15 cm," so it does not match.

15. Answer: Right. In the left code, the robot continues moving when the obstacle is close, so it does not meet the requirement. In the right code, the robot stops when the distance becomes less than 15 cm, so it is the correct answer.

More Relational Operators with Conditionals

Coding Concepts

1. Answer: 100 , True. The block shown is the comparison block (<) in LEGO SPIKE. It checks if the first value is stricktly less than 100, returning True or False.

2. Answer: Left. The left operator checks if the distance is less than 15, which means the obstacle is very close to the robot. The right operator checks for greater than 15, which means far away, so left code is the correct answer.

3. Answer: B. The ultrasonic sensor only measures distance using sound waves, not color, so it cannot detect the color of objects.

4. Answer: C. The ultrasonic sensor can detect walls by measuring distance, helping the robot navigate and find the exit in a maze.

5. Answer: A. The ultrasonic sensor measures distance by sending sound waves forward, so you must point it at the object you want to detect for an accurate reading.

6. Answer: A. You use the "Move Forward" block so the robot can drive closer while the ultrasonic sensor checks the distance.

7. Answer: Right. The right code is correct because it moves forward when the obstacle is less than 15 cm away and plays the meow sound when it is farther than 15 cm. The left code does the opposite, it moves only when the obstacle is more than 15 cm away, so it does not match the description.

8. Answer: Left. The left code is correct because Spike Wally moves forward when someone comes close (distance < 15 cm), which matches being "friendly to strangers." In the right code, Spike Wally moves backward when someone comes close, so it does not show friendly behavior.

9. Answer: Left. The left code is correct because when someone comes close (distance < 15 cm), Spike Wally moves backward, which matches "being shy."
The right code moves forward when someone is close, which does not match shy behavior.

10. Answer: Right. The right code is better because it uses a forever loop, so Spike Wally keeps checking the distance and responds continuously. The left code only checks the distance 10 times, so it will stop working after that.

11. Answer: Same. Both codes are the same because "is closer than 15 cm" means exactly the same as "distance in cm < 15".

12. Answer: Different. Both codes are different because the left code checks if the robot is farther than 15 cm, while the right code checks if it is closer than 15 cm.

13. Answer: B. The robot moves only when an obstacle is closer than 15 cm. It stops in the “else” section, which happens when the obstacle is more than 15 cm away, so the correct answer is B.

14. Answer: D. The code is correct, but the robot is not moving because the motors are most likely not plugged into the default ports A and B. Without a “set movement motors” block, if the motors are connected to other ports, the robot cannot move. So the robot needs the block “set movement motors to E+F” for movement to work.

15. Answer: C. When the obstacle is exactly 15 cm away, it is not less than 15 and also not greater than 15, so both conditions are false. Because neither “distance < 15” nor “distance > 15” is true, the robot goes to the final “else” section and stops moving.

Relational Operators with Sound

Coding Concepts

1. Answer: C. The “More Than” and “Less Than” operators are used to compare numbers or values. A and B are wrong because operators do not compare brick sizes or speed directly. D is wrong because color is detected by the Color Sensor, not by comparison operators.

2. Answer: C. The “More Than” operator is shown in Spike as the greater-than symbol (>), which is the math symbol for comparing values.

3. Answer: A. 5 is more than 3, so the comparison 5 > 3 returns True.

4. Answer: C. “Less Than” (<) is used to check if one number is smaller than another.

5. Answer: B. To check if apples are less than bananas , you use the “Less Than (<)” operator.

6. Answer: A. With a reading of 20 cm, the first condition (distance < 25) is true, so the robot plays beep note 60.

7. Answer: B. With a reading of 30 cm, the first condition (distance < 25) is false, but the second condition (distance < 50) is true, so the robot plays beep note 62.

8. Answer: C. With a reading of 60 cm, both first condition (distance < 25) and second condition (distance < 50) are false. Since neither condition is true, the code goes to the “else” section and plays beep note 64.

9. Answer: A. With a reading of 4 cm, the first condition (distance < 25) is true, so the robot plays beep note 60

10. Answer: B. With a reading of 25 cm, the first condition (distance < 25) is false, but the second condition (distance < 50) is true, so the robot plays beep note 62.

11. Answer: C. With a reading of 100 cm, both first condition (distance < 25) and second condition (distance < 50) are false. Since neither condition is true, the code goes to the “else” section and plays beep note 64.

12.. Answer: C. If the distance is always less than 25 cm, the first condition stays true every time, so the program never reaches the other conditions and keeps playing only the first beep note.

13. Answer: C. With a reading of 20 cm, the first condition (distance > 10) is true, so the robot plays beep note 64.

14. Answer: A. With a reading of 5 cm, both first condition (distance > 10) and second condition (distance > 5) are false. Since neither condition is true, the code goes to the “else” section and plays beep note 60.

15. Answer: C. With a reading of 20 cm, the first condition (distance > 10) is true, so the robot plays beep note 64.

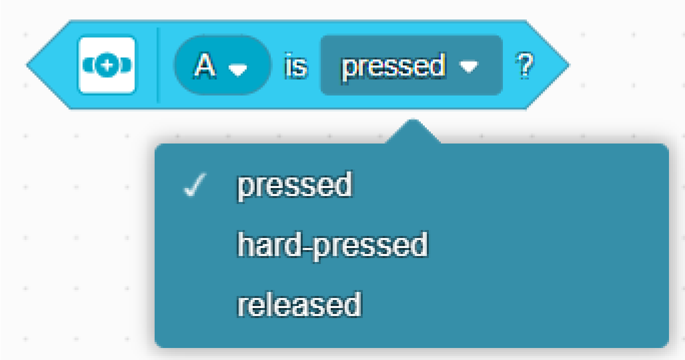
Fractions and Randomness with Conditionals

Coding Concepts

1. Answer: A, pressed. This block checks if the force sensor, connected to port A, has been pressed.

2. Answer: Stops. This block stops the motor connected to port A.

3. Answer: D. “Soft pressed” is not a mode of the touch sensor.



4. Answer: B. The Touch Sensor’s main purpose is to detect when it is pressed or released. Options A, C, and D are not functions of the Touch Sensor.

5. Answer: D. The Touch Sensor works when something physically presses or releases it. The other options describe abilities it does not have.

6. Answer: C. When pressed, the Touch Sensor sends a signal that tells the program it has been touched. The other options are not real sensor outputs.

7. Answer: C. The Touch Sensor is designed to detect forces up to 10 newtons, which is its maximum sensing limit. The other values are either too low or higher than what the sensor can read.

8. Answer: D. The Random Block Operator chooses an option by picking one item at random from a list of choices.

9. Answer: B. The Random Block Operator can choose any color that you include in its list, not just one specific color like blue, green, or red.

10. Answer: B. Each time the Random Block Operator runs, it picks a new random choice, so the robot can perform different actions instead of repeating the same one.

11. Answer: First. The first block, “pick random 1 to 10,” chooses a new random number each time, which can be used to make the robot move unpredictably. The other blocks do not generate random values, so they won’t create random movement.

12. Answer: Right. The right code is better because using seconds creates a truly random spin, while using rotations makes the wheel stop at the same number.

13. Answer: C. Using a random number of rotations still makes the wheel stop at the same number each time.

14. Answer:D. A negative value for time is invalid, so the motor will not spin. Other options use either negative speed or negative rotations, which correctly spin the wheel anti-clockwise.

15. Answer:A. Both programs will behave the same, as pressing the touch sensor makes the robot move randomly 1 to 10 seconds.

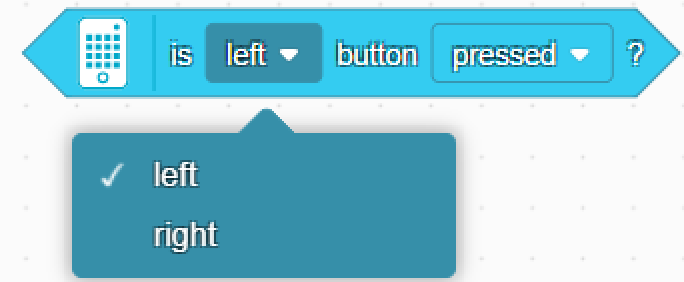
Relational and AND operators with Conditionals

Coding Concepts

1. Answer: And , True. The AND block computes the logical And operation on the two inputs, returning True or False.

2. Answer: C. Ultrasonic sensors detect how far an object is, so you use the “distance” block.

3. Answer: A. The hub has only two buttons — left and right.



4. Answer: B. The “And” operator is used to check two conditions at the same time, and it only becomes true when both conditions are true.

5. Answer: C. The “And” operator is used to check two conditions at the same time.

6. Answer: C. The “And” operator becomes true only when both conditions are true at the same time, so it must be daytime *and* the sun must be shining.

7. Answer: C. With the “And” operator, both conditions must be true; if even one is false, the whole condition becomes false.

8. Answer: C. The “And” operator is used when you want something to happen only if both conditions are true, which matches the requirement in the question.

9. Answer: B. With a reading of 3 cm, the condition “closer than 15 cm” is true, so all blocks inside the “if” section will run: the robot moves up, plays the Drum Roll sound, and then moves down. This means it both moves and makes a drum roll.

10. Answer: C. With a reading of 20 cm, the condition “closer than 15 cm” is false, so none of the blocks inside the “if” section will run. Since there is no “else” block, Spike Drummer simply does nothing.

11. Answer: C. With a reading of 15%, the condition “closer than 15 cm” is false because 15% is not less than 15cm. Since the condition is false and there is no “else” block, Spike Drummer will do nothing.

12. Answer: C. Air has no obstacle, so the sensor reading will be greater than 15 cm, making the condition “closer than 15 cm” false. Since the condition is false and there is no “else” block, Spike Drummer will do nothing.

13. Answer: C. The ultrasonic sensor does not sense anything, so the condition “closer than 15 cm” is false. Because the code uses an AND operator, both conditions must be true, so the whole condition fails and the robot does nothing.

14. Answer: B. With a reading of 5cm, the condition “closer than 15 cm” is true, and the touch sensor is pressed, making both conditions true. Since the AND operator is satisfied, the robot moves and plays the drum roll.

15. Answer: D. There are four conditions connected by AND blocks, meaning all four must be true for the Drum Music to play.

Logic with Conditionals

Coding Concepts

1. Answer: Center, red. This block sets the center button on the Spike hub to light up in red.

2. Answer: C. An “If-Else” statement lets the robot choose one action if the condition is true and a different action if the condition is false, allowing decisions between two pathways.

3. Answer: B. An “If” statement checks one condition and decides what the robot should do based on whether that condition is true or false.

4. Answer: B. You can add as many “If-Else” statements as you want in a program, which means you can check many different conditions one after another.

5. Answer: B. Multiple “If” statements can be stacked one after another in a program, so you can check as many conditions as you want.

6. Answer: D. If none of the conditions for sunny, rainy, or snowy are true, the program will skip all those checks and run the “Else” part, which is meant to handle any situation that doesn’t match the earlier conditions.

7. Answer: B. In an “If-Else” statement, if the first condition is false, the program does not stop; it simply moves on to check the next condition. This allows the program to test multiple possibilities one after another until one becomes true or it reaches the final “else”.

8. Answer: C. In an “If-Else” statement, once the first condition is true, the program immediately runs the code inside that “if” block and does not check the other conditions. The remaining conditions are ignored because the decision has already been made.

9. Answer: D. When separate individual “If” statements are used (not If-Else), the program checks each condition one by one, and every condition that is true will run its code. Since the robot detects red, blue, and green, all three “If” conditions are true, so all three statements will run.

10. Answer: B. If you want the robot to react differently to red, blue, and green, you need one separate “If” statement for each color. Each “If” checks its own condition, so three different colors require three different checks.

11. Answer: B. You have three different conditions to check: morning, afternoon, and evening. Each condition needs its own “If-Else” statement so the robot can respond differently depending on the time. Therefore, you need three separate If-Else statements.

12. Answer: B. A LEGO Spike robot’s color sensor can only read one color at a time.

13. Answer: Right. The right code is better because it uses an “If–Else” structure to check the colors one at a time, so the robot will meow when it sees either red or blue. The left code combines both color checks with an “AND”, which would require the sensor to detect red and blue at the same time, something a single-color sensor cannot do, so it would never meow.

14. Answer: Right. The right code is better because it uses an “If–Else” structure, which checks for red first and then checks for blue only if red is not detected, ensuring the robot makes one clear meow based on the correct color. The left code uses nested “If” blocks, so if the sensor reads colors quickly, the robot might try to meow twice or behave unpredictably.

15. Answer: D. If the camera senses red, it changes the light to red; if it senses blue, it changes the light to blue; but if the color is NOT red and NOT blue, it goes to the last “else” and plays the Meow sound.

Area and Perimeter with Conditionals

Coding Concepts

1. Answer: A, D. The Touch Sensor in LEGO Spike can measure how much it is pressed. It reports this in percentage (%) to show how hard it was pushed, and in Newtons (N) to show the actual force applied.

2. Answer: A. A condition in programming is a statement that can only be True or False. The program checks this True/False result to decide what action to take next.

3. Answer: C. the "Else" part runs only when the If condition is false.

4. Answer: B. When A is pressed, the robot plays the Car Horn sound and skips the rest of the code in the "else" section, even if Button B is also pressed.

5. Answer: A. When you press both buttons at the same time, the first "if" detects that button A is pressed and plays the car horn, and the second "if" also detects that button B is pressed and plays the basketball bounce. Since both conditions are true, the robot plays both sounds.

6. Answer: C. When no buttons are pressed, the condition "A is pressed" is false, so the robot skips the car horn and goes to the "else" block, which plays the basketball bounce sound.

7. Answer: D. A single port can only connect to one device at a time, so if Port A is already used for a motor, the robot cannot detect a button press on the same port, and the program will not work as intended.

8. Answer: B. When you are pressing the button, this condition is false, so the robot follows the "else" part and stops moving.

9. Answer: A. The code has two separate "if" statements, so when both Touch Sensor A and Touch Sensor D are pressed, both conditions become true. This causes the robot to run the start-moving command and the stop-moving command at the same time, making it unclear which action it should follow.

10. Answer: D. When both Touch Sensors A and D are pressed, the robot checks Sensor A first. When the first "If" condition becomes true, the robot immediately runs the start moving command and the "Else" part (which checks Sensor D) is completely skipped.

11. Answer: C. A loop in programming means the computer repeats the same set of instructions again and again until it's told to stop.

12. Answer: C. When only Touch Sensor A is pressed, the first AND condition is not met because B is not pressed. The code then checks the second "if", which detects that A is pressed and runs the block that moves the robot for −1 rotation, meaning it will move backward.

13. Answer: B. When both Touch Sensors A and B are pressed, the first "if" condition becomes true, so the robot runs the block inside it, which tells the robot to move straight for 1 rotation, meaning it will move forward.

14. Answer: B. When only Touch Sensor A is pressed, the first “if” condition becomes true right away, so the robot runs the block inside it, which tells it to move straight for 1 rotation.

15. Answer: B. When both Touch Sensors A and B are pressed at the same time, the first if condition (A is pressed) is already true, so the robot immediately runs the block inside it, which tells it to move straight for 1 rotation and the robot will not check the later conditions.

Arithmetic Sequence with Wait Until

Coding Concepts

1. Answer: A. At the start of the loop, the code immediately sets the motor speed to 100% and runs the motor. When Button B is not pressed, the helicopter stays at the first speed setting, which is 100%.

2. Answer: B. After starting at 100%, the program waits until Button B is pressed once, and when that happens, it changes the motor speed to 80%.

3. Answer: C. After strating at 100%, the program waits for Button B to be pressed once, which changes the speed to 80%, and then waits for Button B to be pressed a second time, which changes the speed to 60%.

4. Answer: A. After the third press of button B, the code reaches the bottom and then loops back to the top, where the speed is reset to 100%.

5. Answer: B. The program steps through each speed block every time B is pressed, so it takes four presses to reach the block that sets the speed to 80% again. After the fourth press, the speed is reset to 80%.

6. Answer: B. The first press sets the speed to 100%, and the second press moves the program to the next block, which sets the speed to 80%. So, it takes 2 presses to reach the 80% speed setting.

7. Answer: A. Each press of button B moves the program to the next speed (100% → 80% → 60%), and after the third press it loops back to the start. The fourth press returns to the block that sets the speed to 100%.

8. Answer: D. The code never includes any block that sets the speed to 0%, so no matter how many times you press Button B, the motor will never change to 0% speed.

9. Answer: A. Each press of Button B moves the program to the next speed block in order. The program sets the speed to 80% after reaching the fourth “wait until B is pressed” block.

10. Answer: B. Each press of Button B moves the program to the next speed block in order. The program sets the speed to 60% after reaching the fifth “wait until B is pressed” block.

11. Answer: C. Each press of button B moves the program to the next speed block in the cycle (100% → 80% → 60% → repeat). So, the 7th press lands on the block that sets the speed to 100% again.

12. Answer: Right. The left code runs the motor once and never checks the button again, but the right code keeps responding every time the button is pressed, so the right code is better.

13. Answer: Left and Right. Even though both codes are arranged slightly differently, the behavior is identical. They check whether Button B is pressed and then set the speed to 100% and start the motor.

14. Answer: B. Without the “Wait 1 Second” block, the program would change the speed instantly and skip through the steps too fast. The robot wouldn’t have time to update its speed properly, so the speed changes would not work as intended.

15. Answer: C. An “If Then” block does not stop the program for 1 second. It simply checks the condition instantly and continues running.