The Lab
Junior Program

Term 4
Home Workbook
Answer Key

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

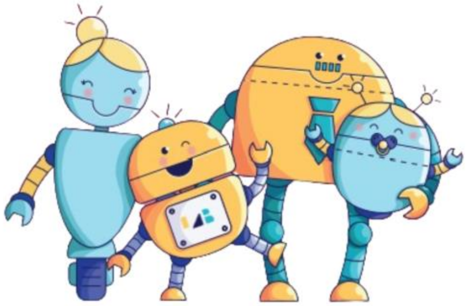
Version Oct 2025

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

Probability with Variables

Coding Concepts

1. Answer: name. Allows you to name a variable.

2. Answer: Counter, 0. Set the name of the variable, Counter, to a value of 0.

3. Answer: change, 1. Allows you to change the value of the variable by 1 , which means it increases the counter by 1 each time the block runs.

4. Answer: B. A variable is like a labeled box that stores data which can change while the program is running, for example, scores or random numbers.

5. Answer: C. Variables let you store and update information (like scores or random choices), which makes your program flexible instead of using fixed numbers everywhere.

6. Answer: Left. To display the velue of the Random Number, you must use the block that shows the value of the variable Random Number on the hub's screen.

7. Answer: A. At the start of the game, the random number should be 0 so that the program has a known starting value before you generate a new random choice.

8. Asnwer: C. The variable 3 is linked to the motor on Port C, which controls the Paper arm, so the Paper motor should be activated.

9. Answer: B. The number 2 is linked to the motor on Port B, which controls the Scissors arm, so the Scissors motor should be activated.

10. Answer: A. The number 1 is linked to the motor on Port A, which controls the Rock arm, so the Rock motor should be activated.

11. Answer: C. The random block picks a whole number from 5 up to 7, so the only possible values the robot can show are 5, 6, or 7.

12. Answer: B. When the random number is 5, the first "if" condition is true, so you will hear a Cat Meow.

13. Answer: C. When the random number is 6, the second "if" condition is true, so you will hear a Basketball Bounce sound.

14. Asnwer: D. When the random number is 7, the thrid "if" condition is true, so you will hear a Car Horn sound.

15. Answer: D. The random number is always between 5 and 7, and each of those numbers has a matching sound, so the robot will never be silent.

Algebra with Variables

Coding Concepts

1. Answer: C. Changing favourite Color from "blue" to "red" shows that a variable can store different values over time; its content is not fixed.

2. Answer: B. Variables are mainly used to store and manage data that can change during the program, such as scores, guesses, or sensor readings.

3. Answer: C. A temperature variable would hold the current temperature in degrees

4. Answer: C. Player Score follows normal naming rules - no symbols at the start, and it describes what it stores so it is a valid variable name

5. Answer: A. When you set counter to 5, the variable stores the number 5, not the word "counter" or a picture.

6. Answer: A. Whenever you give a variable a new value, you are assigning a value to it, which is called variable assignment.

7. Answer: A. The variable should describe what it stores; using the Score for the number of coins collected makes it easy to understand.

8. Answer: D. You want the count to go up by 1 each time the button is pressed, so the touch sensor is best for detecting each press clearly.

9. Answer: C. The wait block gives you time to press and release the touch sensor, so one press is counted once and not many times by accident.

10. Answer: B. When the button is pressed, the program increases the counter by 3 each time. The counter starts at 0, so the values it can display are multiples of 3: 3, 6, 9, 12, ...From the options given, the only possible value is 9.

11. Answer: C. To increase the counter by 15 each press, you must change the variable by 15, so 15 is the number you should input.

12. Answer: B. If the number on the display goes 10, 20, 30, the block is adding 10 each time; the red arrow value must therefore be 10.

13. Answer: D. The robot will only display "Hello" when the counter reaches 100, because that is the condition set in the code.

14. Answer: B. When the button is pressed, the program increases the counter by 2 each time. The counter starts at 0, so the values are multiples of 2: 2, 4, 6, 10, ...From the options given, the only possible value is 4.

15. Answer: C. The motors are activated only when the counter reaches the specified value in the condition, which is 6 in this code.

Range with Variables

Coding Concepts

1. Answer: C. The score variable stores how many coins or points you have collected in the game, not the name or the colour of the game.

2. Answer: C. Setting age to 12 means the variable age currently holds the number 12, but you can still change it later if needed.

3. Answer: C. The robot will display 2 because the counter is first set to 0, then the second “set Counter to 2” block overwrites the earlier value of 0.

4. Answer: A. The Counter is set three times, and the last value you assign is always the one that gets displayed. Here, the code sets Counter to 0, then to 2, and finally back to 0 again, so the robot will show 0 in the end..

5. Answer: A. The counter is first set to 0, then set to 2, but inside the repeat loop it is set back to 0 ten times, so the final value that gets written is 0.

6. Answer: D. The Counter is first set to 0, then set to 2. Inside the repeat loop, the block “change Counter by 1” increases the value by 1 each time, and it runs 10 times. So the Counter goes from 2 to 2 + 10 = 12, and that is the number displayed on the robot.

7. Answer: B. The Counter is first set to 0 and then set to 2. To make the final value equal 4, the Counter must increase from 2 to 4, which means it needs a total increase of 2. Since the loop runs twice, each increase must be 1, so the correct number to input is 1

8. Answer: C. The Counter must reach 10. Since the loop runs twice and adds 4 each time (total +8), the starting value must be 2 (10-8 = 2).

9. Answer: D. After the loop, the program sets the Counter to the number you input. To display 10, you must directly set the Counter to 10.

10. Answer: A. The left button subtracts 1 each time, and the right button adds 1. The Counter is first set to 0. Pressing the left button twice gives: 0 − 1 − 1 = −2. Pressing the right button once adds 1: −2 + 1 = −1. So the final Counter value is **−1**,

11.Answer: B. The left button subtracts 1 each time, and the right button adds 1. The Counter is first set to 0. Pressing the left button 5 times gives: 0 − 5= −5. Pressing the right button 10 times gives: −5 + 10 = 5. So the final Counter value is **5**,

12. Answer: B. This block runs once at the beginning to set up the starting values and begin the main loop, which effectively starts the game.

13. Answer: D. These blocks adjust the Counter variable whenever the left button or right button is pressed, so they are updating the counter value.

14. Answer: B. These blocks show whatever number is stored in the Counter so the player can see their current value on the hub.

15. Answer: A. The highlighted blocks resets the counter back to 0 so the program can start the game again from the beginning.

Time with Variables

Coding Concepts

1. Answer: B. The variable name stores text like "Alice" or "Bob", so it holds a name rather than a number, colour, or sound.

2. Answer: B. A single variable normally has one data type at a time; you don't freely mix different kinds of data inside the same variable.

3. Answer: B. "isRaining" being true or false tells you whether it is raining or not, which is a yes/no (Boolean) type of information.

4. Answer: C. Changing a variable using "if-else" or other logic is called manipulating the variable's value based on conditions.

5. Answer: C. While you can write words like "Happy" as text, emotions themselves are not a standard data type like numbers, text, or dates.

6. Answer: C. The code is using the variable called "Counter" to keep track of how many times the button has been pressed.

7. Answer: A. Waiting 1 second gives time to press and release the button so each press is counted once instead of many times very quickly.

8. Answer: B. A short delay (1 second) helps ensure each button press is counted only once. 0 seconds is incorrect because without any delay, the program would count the same press many times. 100 seconds and 1000 seconds are also incorrect because the delays are far too long, causing the program to freeze after each press and making it impossible to count button presses normally.

9.Answer: A. The display shows 0 because the program writes the current value of "Seconds" to the screen immediately after the wait but before adding 2 to it.

10. Answer: C. Resetting "Seconds to 0" allows the game to start counting again from the beginning for the next round.

11. Answer: B. Picking a random number from 1 to 10 makes the game different each time it starts, which keeps it interesting.

12. Answer: C. The variable "Seconds" increases by 2 every time the A button is pressed. If the value shown is 60, then the robot must have counted 60 ÷ 2 = 30 presses.

13. Answer: C. The variable "Seconds" is misleading. It increases the value by 2 each time the A button is pressed, so it is actually counting button presses, not time. Since the name doesn't match what the variable truly represents, the variable name is incorrect.

14. Answer: C. The variable “Random Number” is set to a negative value while the variable “Seconds” starts at 0, meaning the condition “Seconds > Random Number” is always true, causing the program to reset continuously instead of working as intended.

15. Answer: B. Adding a random block makes the game more exciting and it’s unpredictable, because you can’t easily guess what will happen next.

Ambient Light Intensity with Variables

Coding Concepts

1. Answer: B. A pulley system is used to lift or move heavy objects more easily by changing the direction and sharing the load.

2. Answer: A. A basic pulley has a wheel and a rope or string that goes over it, which helps lift things.

3. Answer: A. The code checks that if the level is greater than 10 and less than 100, so the range is between 10 and 100.

4. Answer: C. The code checks that if the level is greater than -10 and less than 100, so the range is between -10 and 100.

5. Answer: C. The code checks that if the level is greater than -10 and less than 100, so the level is between -10 and 100.

6. Answer: C. The code checks that if the level is greater than -20 and less than 100. Among the options, only 20 is inside the allowed range.

7. Answer: B. The code checks that if the level is greater than 100 or less than -20. Among the options, only -20 fits one of the conditions.

8. Answer: A. If there are only 4 levels, the maximum level number should be 4, so the limit value in the code should be 5 to stop going beyond 4.

9. Answer: Right. The right code is correct for setting a maximum level because it checks whether the Level has reached or exceeded the maximum value using Level < (max). When the Level is too high, the program can stop increasing.

10. Answer: Left. The left code is correct for setting a minimum level because it checks whether the Level is > (min). When the Level becomes too low, the program can stop decreasing.

11. Answer: B. The codes stop the level from going above the highest allowed value, so it sets the maximum level.

12. Answer: A. The wait block gives time to press and release the button properly so one press is not counted many times.

13. Answer: A. The codes stop the level from going below the lowest allowed value, so it sets the minimum level.

14. Answer: B. The motors move clockwise when the left button is pressed and the level is less than 10, matching the conditions in the code.

15. Answer: D. The motors move anti-clockwise when the right button is pressed and level is more than 0, matching the conditions in the code.

Calibration with Variables

Coding Concepts

1. Answer: C. Random events make games more exciting because players cannot predict exactly what will happen next.

2. Answer: C. If the treasures appear in different places each time, players are encouraged to explore and replay the game to see new outcomes.

3. Answer: 0, True. The block checks if a value is between –10 and 10; if it is, the result is True, otherwise it is False.

4. Answer: C. Overlapping ranges mean the same value can satisfy more than one condition, so the program may behave in confusing or unexpected ways.

5. Answer: C. If morning and noon overlap in the code (for example, if 12:00 PM counts as both morning and noon), the program might not know whether to treat the time as morning or noon, causing unclear behaviour.

6. Anwer: D. This creates confusion: Does the player win immediately upon reaching 6 points (satisfying the "more than 5" rule), or are they supposed to wait until they hit 10? The broad rule ("more than 5") makes the specific rule ("10 points") meaningless and confusing, making the actual goal of the game unclear.

7. Answer: D. When two rules cover the same situation (for example, a sunny day is also a day when it is not raining), the program might not know which specific instruction is the priority. This leads to logical errors or unpredictable behavior because the computer cannot "decide" which rule is more important on its own.

8. Answer: C. Clear, non-overlapping conditions help the program act in a predictable way so you know exactly which path the code will follow.

9. Answer: C. The original code uses > -10 and < 10, which exclude the numbers -10 and 10, whereas the "in between" block is inclusive (it includes the start and end numbers); therefore, to cover the same range of integers (from -9 to 9), the boundaries in the block must be set to -9 and 9.

10. Answer: C. The display block shows whatever number is stored in the Battery variable, not just the word itself.

11. Answer: C. The motor speed is set using the value stored in Battery, so if Battery is 60, the motors run at 60% speed.

12. Answer: C. The condition checks if the Battery value is more than 100, and if so, it stops the motor.

13. Asnwer: D. The ranges 40–49 and 45–60 overlap, so numbers like 46 fit both, which makes the code’s behaviour unclear.

14. Answer: B. The ranges 40–49 and 45–60 overlap, the robot will not know what to do when the light intensity is 46%, leaving a gap in the logic.

15. Answer: B. If the battery level increases by increments that skip the number 100 (e.g., jumping from 99 to 101), the exact equality condition will never be met and the motor will fail to stop.

Coordinates with Variables

Coding Concepts

1. Answer: B. A pixel is a tiny coloured dot; many pixels together form the pictures and text you see on the screen.

2. Answer: B. Zooming in too much makes you see the individual pixels, so the picture starts to look blocky instead of smooth.

3. Answer: C. Pixels are the basic building blocks for images and videos; changing their colours creates moving pictures on the screen.

4. Answer: D. The LEGO Spike hub uses a 5 by 5 LED matrix, meaning there are 5 pixels across and 5 pixels down.

5. Answer: D. Based on the standard format for coordinate blocks in coding, the order of the values is (x, y). From the code, the x position is set to 2 and the y position is set to 3, which gives the pixel's coordinate as (2,3).

6. Answer: Left. The left code changes the x value when the left or right button is pressed, so the pixel moves sideways across the screen.

7. Answer: Right. The right code changes the Y value when the left or right button is pressed, so the pixel moves up and down on the screen.

8. Answer: C. The coding blocks make sure x and y stay between 1 and 5 so the pixel cannot move off the 5x5 screen. The top box (Player < 5) allows the variable to increase only if it is currently less than 5. The bottom box (Player > 1) allows the variable to decrease only if it is currently greater than 1.

9. Answer: A. By fixing the player y-axis at 5, the pixel can only move left and right along the bottom row of the screen.

10. Answer: A. The blocks move the falling pixel by changing its y value from top to bottom, making it drop down the screen.

11. Answer: B. The code updates x and y so the pixel's position changes over time as it moves or falls.

12. Answer: C. Using a random x value lets the falling pixel start in different columns, making the game less predictable.

13. Answer: B. The block sets Y to 1 to ensure that every new falling pixel resets to the top of the screen, creating a consistent starting point for the animation before it moves down.

14. Answer: C. Since the LED matrix is 5x5, x and y values must be from 1 to 5, so the random range matches the size of the screen.

15. Answer: C. Repeating 5 times makes the falling pixel move through each of the 5 rows on the 5x5 matrix from top to bottom.

Speed with List/Array

Coding Concepts

1. Answer: A, Newton. A variable containing the current pressure detected by the force sensor in % connected to Port A. The unit of measurement can change between % or Newton.

2. Answer: A. In programming, true means the condition is correct or happening.

3. Answer: C. False means the condition is not correct or not happening, so the code will do the "else" section.

4. Answer: C. If both conditions are true, it means it is daytime and the sun is shining.

5. Answer: B. If the condition "Is the cat asleep?" is false, that means the cat is not asleep, so it must be awake.

6. Answer: A. The correct answer is A because the condition "time less than 5 minutes" being false indicates that at least 5 minutes have passed, implying the game might have ended; this rules out options B and D because a new game would have a time near zero, and eliminates option C as it is a subjective opinion not based on the data.

7. Answer: A. Because the speed is fixed at 40% and never changes, you can simply type 40 directly and you do not need a variable for it.

8. Answer: C. Pressure cannot be negative in this context; you can only press from 0% up to the maximum, so negative values don't make sense.

9. Answer: C. The wait block makes the petrol level drop step by step every second instead of dropping all at once.

10. Answer: A. With the way the variable is changed, the first update makes the value go from 0 down to −1, so −1 is shown first.

11. Answer: D. To start at 100, you must set the Petrol Level variable to 100 at the beginning of the program.

12. Answer: B. The pressure ranges (10–20 and 15–30) overlap between 15 and 20, which causes both code blocks to execute for those specific values and results in unintended double fuel consumption.

13. Answer: A. At pressure 20, the first condition (10-20) and second condition (15-30) are true.

14. Answer: C. At pressure 20, the program meets the first condition but not the second, so the first is true and the second is false.

15. Answer: B. At pressure 30, the program meets the second condition but not the first, so the first condition is false and the second is true.