

TuringLab Smart Cities Mapping (Secondary)

Overview & Mapping – Programme of Study (Computing)

Lesson	Lesson Objectives	Programme of Study for Computing statements
1	Understand <ul style="list-style-type: none"> • what are Smart Cities • In this unit we will use code to control a virtual smart city 	3.1, 3.5
2	Understand <ul style="list-style-type: none"> • how to use <code>print()</code> to output data, including multiple values • What is meant by a 'data type' • The difference between an integer, float, character and boolean • What is meant by a variable and why they are used • The difference between declaration and assignment Use <ul style="list-style-type: none"> • A variable to output data • the <code>get_time()</code> function to return the current time in the simulation • the <code>get_dark()</code> function to return a boolean (i.e - True if it is dark) 	3.3, 3.6, 3.8
3	Understand <ul style="list-style-type: none"> • The programming construct selection • That conditional statements can be used to control the flow of execution • That comparison operators compare two values • That logical operators modify or join conditions Use <ul style="list-style-type: none"> • The <code>==</code> and <code>!=</code> comparison operators and the <code>not</code> logical operator within basic if-statements • Variable assignment • The <code>turn_on</code> and <code>turn_off</code> commands to control the streetlamps • 	3.3, 3.4, 3.6
4	Understand <ul style="list-style-type: none"> • Basic equality operations <code>==</code> and <code>!=</code> • How conditions evaluate to boolean values 	3.1, 3.3, 3.4

CURRICULUM MAPPING | Turing Lab – Smart Cities (Secondary)

	<p>Identify</p> <ul style="list-style-type: none"> • Different data types used within an algorithm • Variables used within an algorithm <p>Use</p> <ul style="list-style-type: none"> • equality operations within a conditional statement • <code>`print()`</code> to output multiple values to the console • Correct syntax when writing selection algorithms <p>Create</p> <ul style="list-style-type: none"> • An algorithm using variable assignment and conditional statements 	
5	Create an algorithm from scratch using conditions, variable assignment and basic operators to light up the dark	3.1, 3.3
6	Create an algorithm from scratch using conditions, variable assignment and basic operators to turn on the lights	3.1, 3.3
7	<p>Understand</p> <ul style="list-style-type: none"> • additional comparison operators: <code><</code>, <code>></code>, <code><=</code>, <code>>=</code> • additional logical operators: <code>or</code>, <code>and</code> • that a constant variable cannot be changed • that appropriate naming conventions help create self-documenting, maintainable code • that a truth table can be used to evaluate complex conditions <p>Identify</p> <ul style="list-style-type: none"> • different operator types • more maintainable code <p>Use</p> <ul style="list-style-type: none"> • conditions combining comparison and logical operators • the <code>`get_wind()`</code> function to fetch the wind intensity in knots • the <code>`set_pitch()`</code> function to set the angle of turbine blades 	3.3, 3.4, 3.6
8	<p>Understand</p> <ul style="list-style-type: none"> • how to use <code>if-else</code> and <code>if-elif-else</code> statements for more powerful selection • that conditions can be nested within one another <p>Use</p> <ul style="list-style-type: none"> • the <code>`low_power()`</code> function to set energy consumers into low power mode 	3.3, 3.6

CURRICULUM MAPPING | Turing Lab – Smart Cities (Secondary)

9	<p>Understand</p> <ul style="list-style-type: none"> • what operators are and what they do <p>Identify</p> <ul style="list-style-type: none"> • logical and comparison operators • equivalent conditions using different operators <p>Use</p> <ul style="list-style-type: none"> • complex combined comparison and logical operations <p>Create</p> <ul style="list-style-type: none"> • an algorithm from scratch using complex conditional statements 	3.3, 3.4
10	Create an algorithm from scratch that uses complex conditional statements to control the streetlamps on-state	3.1, 3.2, 3.3, 3.4
11	Create an algorithm from scratch that uses complex conditional statements and combined operations	3.1, 3.2, 3.3, 3.4
12	<p>Understand</p> <ul style="list-style-type: none"> • counting from zero • what is meant by the term 'index' within loops • that the index variable, i, can be given alternate labels • that 'DRY coding' means 'Don't Repeat yourself' • how to write DRY code using loops to repeat sections <p>Identify</p> <ul style="list-style-type: none"> • the output for algorithms involving loops and conditions • different looping algorithms that produce the same result <p>Use</p> <ul style="list-style-type: none"> • the index, i, to output the current iteration to the console • conditions and loops within algorithms • the `turn_on_by_id`, `turn_off_by_id` and `low_power_by_id` functions to control specific items • the `get_streetlamps` and `get_houses` functions to return the number of each on a street in the simulation 	3.1, 3.3, 3.4, 3.6
13	<p>Identify</p> <ul style="list-style-type: none"> • the correct index within a range counting from 0 • the output of algorithms using loops and index controlled functions 	3.1, 3.3, 3.4, 3.6

CURRICULUM MAPPING | Turing Lab – Smart Cities (Secondary)

	<ul style="list-style-type: none"> • correct algorithms involving loops and conditions given the expected result <p>Use</p> <ul style="list-style-type: none"> • loops and conditions to control multiple objects within a scene by index • arrow key interaction to control characters in the smart city • the function `get_proximity_by_id` to return the distance to the nearest person 	
14	<p>Understand</p> <ul style="list-style-type: none"> • how to combine conditions using logical and comparison operators <p>Identify</p> <ul style="list-style-type: none"> • the output of algorithms containing loops and conditions • equivalent algorithms using nested and non-nested conditions • variables within an algorithm <p>Use</p> <ul style="list-style-type: none"> • a for loop to control specific objects • the correct syntax to for algorithms containing loops and conditions <p>Create</p> <ul style="list-style-type: none"> • algorithms using loops and conditions to solve a task 	3.1, 3.3, 3.4, 3.6
15	Control individual porch lights to respond to inhabitant proximity	3.1, 3.2, 3.3, 3.4, 3.6
16	Control street lamps in the city based upon inhabitant proximity and data from light sensors	3.1, 3.2, 3.3, 3.4, 3.6

Overview & Mapping – Teach Computing Curriculum Year 7 Programming Unit

Label	Teach Computing Curriculum Statement	Covered in Turing Lab – Farm Bot	Farm Bot
-------	--------------------------------------	----------------------------------	----------

CURRICULUM MAPPING | Turing Lab – Smart Cities (Secondary)

PS	Use an IDE to write and execute a Python program.	L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16	
PS	Locate and correct common syntax errors.	L2, L3, L4, L7, L8, L9, L12, L13, L14	
CS	Call functions and use the results they return in expressions.	L1, L2, L3, L4, L5, L5, L6, L7, L8, L9, L10, L11, L12	
PS	Use variables to keep track of information.	L2, L3, L4, L5, L5, L6, L7, L8, L9, L10, L11, L12	
PS	Trace through branches and loops and sketch state.	N/A	
CS	Use selection (if) to control the flow of program execution.	L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16	
DTAS	Create lists and access individual elements	N/A	
PS	Trace through programs that manipulate lists.	N/A	
DTAS	Perform common operations on lists.	N/A	
DTAS	Access individual string elements (characters).	N/A	
CS	Use iteration (while) to control the flow of program execution.	L13, L14, L15, L16	
DTAS	Perform common operations on strings.	N/A	
PS	Use variables to keep track of counts.	L13, L14, L15, L16	
PS	Trace through programs that iterate over sequences using for.	L13, L14	
CS	Use iteration (for) to iterate over lists.	N/A	
PS	Use variables to keep track of sums.	L13, L14, L15, L16	
PS	Combine features to develop solutions to meaningful problems.	L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16	

CURRICULUM MAPPING | Turing Lab – Smart Cities (Secondary)

CS	Use iteration (for) to iterate over strings.	N/A	
	N/A	L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16	Create Functional Algorithms
	N/A	L7, L8, L9, L10, L11, L12, L13, L14, L15, L16	Learn to create more complex conditions by combining comparison and logical operators. Use truth tables, constants and
	N/A	L8, L9, L10, L11, L12, L13, L14, L15, L16	Learn to use nesting and two new conditional statements alongside complex combined conditions within your algorithms.
	N/A	L13, L14, L15, L16	Work through different examples combining loops, conditions, index use and functions to gain greater control of your code.
	N/A	L15, L16	Code an algorithm from scratch using loops and conditions to control the city

PS = Programming Skills

CS = Control Structures

DTAS = Data Types and Structures