

TuringLab Chat Bot 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"> • Explain the concept of a chatbot and its role in our lives • The basic principle behind the operation of chatbots Identify <ul style="list-style-type: none"> • Examples of chatbots in everyday situations • The potential social good applications of chatbots | 3.1, 3.7, 3.8, 3.9 |
| 2 | Understand <ul style="list-style-type: none"> • Output and input in Python using 'print()' and 'input()' • How to use modules • How to define python functions Use <ul style="list-style-type: none"> • Python's input 'print()', 'input()' functions to create chatbot interactions Create <ul style="list-style-type: none"> • Python functions to perform specific tasks Modify <ul style="list-style-type: none"> • Code to incorporate user inputs and provide custom responses | 3.3, 3.4 |
| 3 | Understand <ul style="list-style-type: none"> • Flow of control in a chatbot program • The use of arguments and parameters in functions • The return keyword in functions • The concepts of Sequence, Selection, Iteration, and Abstraction in computational thinking Use <ul style="list-style-type: none"> • Arguments and parameters in function calls • Control structures to manage the flow of a program • The return keyword to obtain values from functions • Chatbot functions such as cprint() and cinput() for interaction Modify <ul style="list-style-type: none"> • Existing function code to enhance chatbot interactions | 3.3, 3.4, 3.6 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | |
|---|--|--------------------|
| | <ul style="list-style-type: none"> • Code to utilise returned values from functions <p>Create</p> <ul style="list-style-type: none"> • Functions to enhance chatbot interactivity and user experience • Code that applies Sequence, Selection, Iteration, and Abstraction concepts | |
| 4 | <p>Understand</p> <ul style="list-style-type: none"> • How to create and call functions • How to use modules in Python • The concept of lists and list indexing • The use of conditional statements • The role of random number generation <p>Use</p> <ul style="list-style-type: none"> • Functions, lists, and modules to create a fact-telling chatbot • Conditional statements to handle user responses • The randint function for random number generation <p>Create</p> <ul style="list-style-type: none"> • A chatbot function that interacts with the user • Multiple functions to handle different aspects of the chatbot <p>Modify</p> <ul style="list-style-type: none"> • Code to add more facts and enhance user interaction" | 3.3, 3.4, 3.6 |
| 5 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of defensive design and its importance • The principle of decomposition • Validation checks in user input handling • How to use Python's lower() and upper() methods <p>Use</p> <ul style="list-style-type: none"> • The concepts of defensive design and decomposition to create an intelligent chatbot • Validation checks to handle user inputs efficiently • Python's string methods like lower() to validate and process user input <p>Create</p> <ul style="list-style-type: none"> • Functions to validate user input and provide intelligent responses • A more robust factbot using principles of defensive design <p>Modify</p> <ul style="list-style-type: none"> • Your existing chatbot code to incorporate defensive design principles and improved user input handling | 3.3, 3.4, 3.6, 3.9 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | |
|---|---|------------------------------|
| 6 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of lists and list indexing • The use of for loops to iterate over lists • The concept of 'Pythonic' code • Various list methods like append(), insert(), pop(), len() <p>Use</p> <ul style="list-style-type: none"> • The 'for' loop to ask multiple questions from a list • List indexing and list methods to manipulate lists • Pythonic way of looping for readability and maintainability <p>Create</p> <ul style="list-style-type: none"> • A list of questions and a function to ask questions • A list to store user responses <p>Modify</p> <ul style="list-style-type: none"> • List contents dynamically through user input • Code to work with lists of any length | 3.3, 3.4, 3.6 |
| 7 | <p>Understand</p> <ul style="list-style-type: none"> • The importance of abstraction in programming • How to cast different data types using `int()`, `float()`, and `str()` • The use of arithmetic operators <p>Use</p> <ul style="list-style-type: none"> • The concepts of abstraction and casting to write reusable code • Arithmetic operators for numerical calculations • Lists and loops to iterate over elements <p>Create</p> <ul style="list-style-type: none"> • A chatbot that calculates carbon footprint based on user's input <p>Modify</p> <ul style="list-style-type: none"> • Code to incorporate more environmental factors in the carbon calculation | 3.1, 3.3, 3.4, 3.6, 3.7, 3.8 |
| 8 | <p>Assess</p> <ul style="list-style-type: none"> • The use of arithmetic operators • The concept of function parameters • The principles of Pythonic list iteration • The concept of decomposition • Type casting in Python • Fixing bugs and adding functionality | 3.1, 3.2, 3.3, 3.4, 3.6 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | |
|----|--|--------------------|
| 9 | <p>Use</p> <ul style="list-style-type: none"> • Variables to store and manipulate user inputs • Arithmetic Operators to calculate carbon emissions • Lists to organize data and List Indexing to access specific items • Loops to run repetitive tasks • Functions to encapsulate specific tasks and reduce repetition in your code | 3.3, 3.4, 3.6, 3.8 |
| 10 | <p>Use</p> <ul style="list-style-type: none"> • The functions 'input()', 'print()' to interact with the user • Functions imported from modules • Loops to ask multiple questions • Conditional statements to process user responses • Lists to store multiple items • User-defined functions to perform specific tasks | 3.3, 3.4, 3.6 |
| 11 | <p>Use</p> <ul style="list-style-type: none"> • Modules and imported functions to create a Carbon Calculator bot • Functions 'round()' and 'print()' for precise value output • Arithmetic operators to calculate carbon footprint • Combining function flows to create a chatbot • Casting and string concatenation to format output messages • Intelligent design to provide custom responses and handle invalid inputs | 3.3, 3.4, 3.6, 3.8 |
| 12 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of string concatenation • The idea of string slicing to create substrings • The usage of negative indexing in string slicing • The necessity of casting integers to strings for concatenation <p>Use</p> <ul style="list-style-type: none"> • The 'input()' and 'print()' functions for interaction • String concatenation and slicing to manipulate data • Custom functions to handle specific string operations <p>Create</p> <ul style="list-style-type: none"> • Custom string manipulation functions 'left()' and 'right()' • Usernames based on specific conditions <p>Modify</p> <ul style="list-style-type: none"> • The output messages based on user input and string manipulations | 3.3, 3.6 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | |
|----|---|------------------------------|
| 13 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of while loops and how they control flow of code • How to use while loops for input validation • The meaning of 'flags' and their usage in breaking out of loops • The concept of 'do-while' loops and their usage • How to check data types using 'isinstance()' <p>Use</p> <ul style="list-style-type: none"> • While loops to control repetition of certain tasks based on conditions • Flags in while loops to control flow • isinstance() to check data types <p>Create</p> <ul style="list-style-type: none"> • Programs using while loops for various tasks, including password checks • Modify • Existing code to add while loops and enhance functionality | 3.1, 3.2, 3.3, 3.4, 3.6 |
| 14 | <p>Use</p> <ul style="list-style-type: none"> • The 'input()' and 'print()' functions to interact with users • String slicing and concatenation to modify strings • Conditional statements to apply different levels of security • The 'import' statement to bring in functionality from a module • Data type casting | 3.1, 3.2, 3.3, 3.4, 3.6, 3.8 |
| 15 | <p>Use</p> <ul style="list-style-type: none"> • The function 'input()' to ask the user for a password • Conditionals to check for different requirements in the password • Built-in string methods such as 'lower()' and 'isdigit()' • The 'len()' function to check password length • Loops to continually ask for input until requirements are met | 3.1, 3.2, 3.3, 3.4, 3.6, 3.8 |
| 16 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of one and two-dimensional lists in Python • How to access items from a list • The usage of list methods such as 'append()' • The idea of updating and assigning values in lists <p>Use</p> <ul style="list-style-type: none"> • One-dimensional and two-dimensional lists to store and organise data <p>Create</p> <ul style="list-style-type: none"> • Empty lists and add items to them | 3.3, 3.6 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | |
|----|--|--------------------|
| | <ul style="list-style-type: none"> • Custom 2D lists to represent given data of real-world scenarios | |
| 17 | <p>Understand</p> <ul style="list-style-type: none"> • The concept of two-dimensional lists and how to access them • How to use nested loops • The use of 'len()' function for list length • The concept of data filtering <p>Use</p> <ul style="list-style-type: none"> • Two-dimensional lists to represent tabular data • Nested loops to access and manipulate two-dimensional lists • Conditional statements to filter data <p>Create</p> <ul style="list-style-type: none"> • Code that correctly navigates through two-dimensional lists • Code that filters data based on specific conditions | 3.3, 3.4, 3.6, 3.9 |
| 18 | <p>Understand</p> <ul style="list-style-type: none"> • The concept and creation of dictionaries in Python • The concept of key-value pairs • How to access and modify data in dictionaries • The idea of nested dictionaries and combining data structures • How to add and delete data in dictionaries <p>Use</p> <ul style="list-style-type: none"> • Dictionaries to store and manipulate information • Square bracket notation for accessing, adding and deleting data • `for` loops to iterate over dictionaries <p>Create</p> <ul style="list-style-type: none"> • Dictionaries to store different types of data • Nested dictionaries for more complex data structure <p>Modify</p> <ul style="list-style-type: none"> • Dictionaries to add and remove data • `for` loops to access data within dictionaries | 3.3, 3.6 |
| 19 | <p>Use</p> <ul style="list-style-type: none"> • Two-dimensional lists to store and retrieve data • Loops to traverse 2D lists and perform operations on each element • The arithmetic operation of addition to sum up values | 3.1, 3.3, 3.4, 3.6 |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

Overview & Mapping – Teach Computing Curriculum Year 7 Programming Unit

| Label | Teach Computing Curriculum Statement | Covered in Turing Lab –Chat Bot | Chat Bot |
|-------|--|--|----------|
| 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, L17, L18, L19 | |
| PS | Locate and correct common syntax errors. | L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| CS | Call functions and use the results they return in expressions. | L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| PS | Use variables to keep track of information. | L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| PS | Trace through branches and loops and sketch state. | L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| CS | Use selection (if) to control the flow of program execution. | L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| DTAS | Create lists and access individual elements | L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| PS | Trace through programs that manipulate lists. | L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| DTAS | Perform common operations on lists. | L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| DTAS | Access individual string elements (characters). | L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | | |
|------|---|--|--|
| CS | Use iteration (while) to control the flow of program execution. | L13, L14, L15, L16, L17, L18, L19 | |
| DTAS | Perform common operations on strings. | L12, L17, L18, L19 | |
| PS | Use variables to keep track of counts. | N/A | |
| PS | Trace through programs that iterate over sequences using for. | L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L12, L13, L14, L15, L16, L17, L18, L19 | |
| CS | Use iteration (for) to iterate over lists. | L16, L17, L18, L19 | |
| PS | Use variables to keep track of sums. | N/A | |
| PS | Combine features to develop solutions to meaningful problems. | L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | |
| CS | Use iteration (for) to iterate over strings. | L16, L17, L18, L19 | |
| | N/A | L1, L2, L3, L4, L5, L6, L7, L8, L9, L10, L11, L12, L13, L14, L15, L16, L17, L18, L19 | Understand what chatbots are, where we encounter them in our daily lives, and set the stage for building your own social good chatbot. |
| | N/A | L3 | Looking at flow of control and flowcharts alongside using functions with return values and parameters |
| | N/A | L5 | Use of defensive design principles, and implement input validation for seamless interaction. |
| | N/A | L13, L14, L15, L16, L17, L18, L19 | Use while loops for input validation, password checks and controlling repetitions in your program. |
| | N/A | L10, L11, L12 | Use the 'cinput()' and 'cprint()' functions to interact with users |

CURRICULUM MAPPING | Turing Lab – Chat Bot (Secondary)

| | | | |
|--|-----|----------|--|
| | N/A | L17, L19 | Navigate through two-dimensional lists using nested loops, access and filter data, and use this data to create things like maps. |
| | N/A | L18, L19 | Get to know dictionaries |

PS = Programming Skills

CS = Control Structures

DTAS = Data Types and Structures