**Main Code**

Open the “data.txt” file in read mode as *file*

Initialise *data* set to the JSON data in *file* as a list of dictionaries

Close *file*

Print “Welcome to the Admin Interface of The Boardgame Catalogue.”

Loop indefinitely

Print “Choose [a]dd, [l]ist, [s]earch, [v]iew, [d]elete or [q]uit.”

Get user input and save as *inp*

Set *inp* to *inp* in all lowercase

Strip all whitespace from *inp*

If *inp* is “a”

Initialise *new\_data* as an empty dictionary

Set *new\_data*[‘name’] to the return of **input\_something** passing in “Enter boardgame name: ”

Set *new\_data*[‘year’] to the return of **input\_int** passing in “Enter release year: ”, 0, and the current year

Set *new\_data*[‘desc’] to the return of **input\_something** passing in “Enter short description: ”

Set *new\_data*[‘players’] to the return of **input\_range** passing in “Enter number of players as a range e.g. 1-4: ”

Set *new\_data*[‘playtime] to the return of **input\_range** passing in “Enter playtime in minutes as a range e.g. 15-30: ”

Set *new\_data*[‘min\_age’] to the return of **input\_int** passing in “Enter the minimum recommended playing age: ”, and 0

Set *new\_data*[‘complexity’] to the return of **input\_int** passing in “Enter complexity(1-5): ”, 1, and 5

Append *new\_data* to *data*

Print *new\_data*[‘name’] + “ added”

Call **save\_data** passing in *data*

Else if *inp* is “l”

If *data* is empty

Print “No boardgames saved”

Restart the loop

Print “List of Boardgames”

Loop for the length of *data* saving the enumeration number as *i*

Print (*i* +1) + “) ” *data*[*i*][‘name’] + “ (” + *data*[*i*][‘year’] + “)”

Else if *inp* is “s”

If *data* is empty

Print “No boardgames saved”

Restart the loop

Initialise *search\_term* as the return of **input\_something** passing in “Enter a search term: ”

Set *search\_term* to *search\_term* in all lowercase

Initialise *search\_results* as an empty list

Loop for the length of *data* saving the enumeration number as *i*

If *data*[*i*][‘name’] or *data*[*i*][‘desc’] contains *search\_term*

Initialise *lst* to a lost containing *i* and *data*[*i*]

Append *lst* to the end of *search\_results*

If *search\_results* is an empty list

Print “No results found”

Restart the loop

Print “Search results: ”

Loop for the length of *search\_results* saving the enumeration number as *i*

Print (*search\_results*[*i*][0] + 1) + “) ” + *search\_results*[*i*][1][‘name’] + “ (” + *search\_results*[*i*][1][‘year’] + “)”

Else if *inp* is “v”

If data is empty

Print “No boardgames saved”

Restart the loop

Initialise *num* as the return of **input\_int** passing in “Boardgame number to view: ”, 1, and the length of *data*

Set *num to num* – 1

Print *data*[*num*][‘name’] + “ (” + *data*[*num*][‘year’] + “)”

Print *data*[*num*][‘desc’]

Print “Players: ” + *data*[*num*][‘players’][0] + “-” + *data*[*num*][‘players’][1]

Print “Playtime: ” + *data*[*num*][‘playtime’][0] + “-” + *data*[*num*][‘playtime’][1] + “ minutes”

Print “Age: ” + *data*[*num*][‘min\_age’] + “+”

Print “Complexity: ” + *data*[*num*][‘complexity’] + “/5”

Else if *inp* is “d”

If *data* is empty

Print “No boardgames saved”

Restart the loop

Initialise *num* to the return of **input\_int** passing in “Enter boardgame number to delete: ”, 1, and the length of *data*

Set *num* to *num* – 1

Delete the item at index *num* from *data*

Print “Deleted boardgame”

Call **save\_data** passing in *data*

Else if *inp* is “q”

Print “Goodbye”

Print “Press enter to close the program”

Wait till the user presses enter

Break out of the loop

Else

Print “Invalid Choice. Please try again.”

**Functions**

Function **input\_something** with parameter *prompt*

Loop forever

Print *prompt*

Get user input and save is as *inp*

Set *inp* to *inp* with all whitespace from the end removed

If *inp* contains any value

Return *inp*

Print “Sorry, you didn’t seem to input anything there. Please try again”

Function **input\_int** with parameters *prompt*, *min\_value*¸ *max\_value*

Loop forever

Call **input\_something** passing in *prompt*

Save the return as *inp*

If *inp* is not an integer

Print “You need to input an integer (a whole number)”

Restart the loop

Set *inp* to be an integer

If *min\_value* had a value passed into the function

If *inp* is less than *min\_value*

Print “The input cannot be less than ” + *min\_value*

Restart the loop

If *max\_value* had a value passed into the function

If *inp* is greater than *max\_value*

Print “The input cannot be more than ” + *max\_value*

Restart the loop

Return *inp*

Function **input\_range** with parameter *prompt*

Loop forever

Print *prompt*

Get user input and save as *inp*

Set *inp* to be *inp* with all whitespace removed

If *inp* does not have a ‘-’ somewhere in it

Print “You need to input a range separated with a ‘-’”

Restart the loop

If *inp* has more than one ‘-’ in it

Print “You can only have one ‘-’, so no negative numbers”

Restart the loop

Split *inp* up into two numbers along the ‘-’ and save them as *num1* and *num2*

If either *num1* or *num2* are not integers

Print “You must input two integers on either side of the ‘-’”

Restart the loop

If either *num1* or *num2* are less than or equal to 0

Print “You must input positive numbers greater than zero”

Restart the loop

If *num2* is less than *num1*

Print “The second number must be greater than the first number”

Restart the loop

Return list of two elements: *num1*, and *num2*

Function **save\_data** with parameter *data*

Open the “data.txt” file in write mode as *file*

Delete all data already within *file*

Write *data* into *file*

Close *file*