**Program #1: Calculate Rainfall Average**

Problem Description: Display total months for which rainfall has been recorded and calculate average rainfall per month.

Pseudo Code:

Prompt the user to input number of years

Validate the user input, ensure value entered is a (+) integer

Loop from 1 to number of years \* 12:

Prompt the user to enter each month rainfall

Store the input in an array/list

Display total number of months: array.size

Calculate sum of all elements in the array: sum(array)

Evaluate average rainfall: sum(array)/ array.size

**GitHub link:**

<https://github.com/sahiraj15/aiml/blob/main/csc500/m5_1_calculate_average_rainfall.py>

**Results:**

**Use case #1**: Enter the rainfall per each month

Inputs:

|  |  |  |
| --- | --- | --- |
| Variable Name | Input Value | Result |
| no\_of\_years | 2 (24 Months) | **Pass** |
| monthly\_rainfall | 8  11  0  1.2  2.2  3.5  12  15  17  18  6  4  8.2  11.4  0.8  3.2  7.2  3.5  5  11  22  14  16  4 |

A screenshot of a computer

Description automatically generated

**Use case #2**: Entering prices in decimals (0s)

Inputs:

|  |  |  |
| --- | --- | --- |
| Item Name | Item Price | Result |
| no\_of\_years | < 1 or a Decimal number | **Fail** |
| monthly\_rainfall |  |

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Challenges encountered and Learning:**

* NumPy functions and apply them on the code
* Converting a list into an array using NumPy
* Expanding on list functions and apply them on the code
* Truncating a decimal number to extract integer value
* Handling errors on user inputs for decimals and negative inputs

**References:**

[**www.realpython.com**](http://www.realpython.com)

[**www.python.org**](http://www.python.org)

**Program #2: Assign reward points on purchased items**

Problem Description: Display items purchased and awarded loyal points based on the number of items.

Pseudo Code:

Prompt the user to enter the number of items purchased

Validate the user input, ensure value entered is a (+) integer

Print Error Info if the input is Invalid

Apply condition for loyal points:

if purchased\_quantity < 2:

number\_of\_loyal\_points = 0

elif purchased\_quantity < 4:

number\_of\_loyal\_points = 5

elif purchased\_quantity < 6:

number\_of\_loyal\_points = 15

elif purchased\_quantity < 8:

number\_of\_loyal\_points = 30

else:

number\_of\_loyal\_points = 60

Manipulate print string for 1 or many books

Print number of books purchased

Print number of loyal points awarded

**GitHub link:**

<https://github.com/sahiraj15/aiml/blob/main/csc500/m5_2_loyal_points_on_purchase.py>

**Results:**

**Use case #1**: Purchased quantity = 1

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 1 | 0 | **Pass** |

A screenshot of a computer program

Description automatically generated

**Use case #2**: Purchased quantity = 3

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 3 | 5 | **Pass** |

A screenshot of a computer program

Description automatically generated

**Use case #3**: Purchased quantity = 6

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 6 | 30 | **Pass** |

A screenshot of a computer program

Description automatically generated

**Use case #4**: Purchased quantity = 10

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 10 | 60 | **Pass** |

A screenshot of a computer

Description automatically generated

**Use case #5**: Purchased quantity = 100

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 100 | 60 | **Pass** |

A screenshot of a computer program

Description automatically generated

**Use case #6**: Purchased quantity = 0

Inputs:

|  |  |  |
| --- | --- | --- |
| Quantity | Loyal Points | Result |
| 0 | 0 | **Pass** |

A screenshot of a computer program

Description automatically generated

**Challenges encountered and Learning:**

* Nested if conditions
* Handling string for singular or plural books for print statements
* Condition design based on the requirements
* Handling errors on user inputs for decimals and negative inputs

**References:**

* [**www.realpython.com**](http://www.realpython.com)
* [**www.python.org**](http://www.python.org)