# Code / Pseudocode

## Part 1

# Get number of years

def get\_years():

years = input("Please enter the number of years of data you will provide and press enter: ")

# guard for invalid entries

try:

if not years.isnumeric():

print("\nPlease enter an integer.")

return get\_years()

years = int(years)

except:

print("\nPlease enter an integer.")

return get\_years()

return years

# Get rainfall

def get\_rainfall(year, month):

rainfall = input("Please enter rainfall (inches) for Year %d %s and press enter: " % (year, month))

# guard for invalid entries

try:

if not rainfall.isnumeric():

print("\nPlease enter a number.")

return get\_rainfall(year, month)

rainfall = float(rainfall)

except:

print("\nPlease enter a number.")

return get\_rainfall(year, month)

return rainfall

months = [

"January",

"February",

"March",

"April",

"May",

"June",

"July",

"August",

"September",

"October",

"November",

"December" ]

# init rainfall list

rainfall = []

# First loop (years)

# get years

num\_years = get\_years()

for i in range(num\_years):

# Second loop (months)

for month in months:

rainfall.append(get\_rainfall(i+1, month))

print("\nTotal number of months of data collected: %d" % len(rainfall))

print("\nThe average rainfall in inches per month for this period was: %0.2f in\n" % (sum(rainfall)/len(rainfall)))

## Part 2

# Get number of books

def get\_books():

num\_books = input("\nPlease enter the number of books you have purchased this month and press enter: ")

# guard for invalid entries

try:

if not num\_books.isnumeric():

print("\nPlease enter an integer.")

return get\_books()

num\_books = int(num\_books)

except:

print("\nPlease enter an integer.")

return get\_books()

return num\_books

# Init rewards values

points\_categories = {0:0, 2:5, 4:15, 6:30, 8:60}

# Init rewards to 0

rewards\_pts = 0

# Get user book purchases

num\_books\_this\_month = get\_books()

# Iterate through rewards levels

for tier in points\_categories:

# if books purchased is greater or equal to current tier set corresponding rewards points

if num\_books\_this\_month >= tier:

rewards\_pts = points\_categories[tier]

else:

break

if rewards\_pts == 0:

print("\nPlease purchase at least 2 books a month to receive rewards points!")

else:

print("\nCongratulations! You've received %d rewards points this month!" % rewards\_pts)

# Application Screenshot

## Part 1

A screenshot of a computer

Description automatically generated

## Part 2

A computer screen shot of a program

Description automatically generated