**Module5 Critical Thinking Git Repo**

|  |
| --- |
|  |

**Part1 source code**

**Problem statement:** Write a program that uses nested loops to collect data and calculate the average rainfall over a period of years.

The program should first ask for the number of years. The outer loop will iterate once for each year. The inner loop will iterate twelve times, once for each month.

Each iteration of the inner loop will ask the user for the inches of rainfall for that month. After all iterations, the program should display the number of months, the total inches of rainfall, and the average rainfall per month for the entire period.

**Code:**

|  |
| --- |
| *from enum import Enum*  *class Months(Enum):*  *JAN =1*  *FEB =2*  *MAR =3*  *APR =4*  *MAY =5*  *JUN =6*  *JUL=7*  *AUG=8*  *SEP=9*  *OCT=10*  *NOV=11*  *DEC=12*    *class Rainfall:*  *num\_years: int*    *def \_\_init\_\_(self, num\_years: int):*  *self.num\_years = num\_years*    *def cal\_avg\_ranfall(self, monthly\_data: dict):*  *total\_months = self.num\_years \* len(Months)*    *total\_rain\_fall = 0*  *for year, monthly\_data in monthly\_data.items():*  *total\_rain\_fall += sum(monthly\_data.values())*    *avg\_rain\_fall = round(total\_rain\_fall / total\_months , 2)*  *print(f"Total number of months: {total\_months}")*  *print(f"Total inches of rain: {total\_rain\_fall}")*  *print(f"Average rain fall per month: {avg\_rain\_fall}")*      *def load\_rainfall\_data(self) -> dict:*  *user\_data = {}*    *for year in range(1, (self.num\_years+1)):*  *user\_data[year] = {}*  *for month in Months:*  *user\_input = input(f"Enter rain fall for year: {year} {month.name} month: ")*  *if (user\_input != None and user\_input != ''):*  *rain\_fall = round(float(user\_input), 2)*  *user\_data[year][month.name] = rain\_fall*  *else:*  *user\_data[year][month.name] = 0*      *return user\_data*    *if \_\_name\_\_ == "\_\_main\_\_":*    *print("Program to calculate the average rain fall, please enter the few years of data")*    *num\_year = int(input("Please enter the number of years: "))*  *rainfall\_calc = Rainfall(num\_years=(num\_year))*    *monthly\_data = rainfall\_calc.load\_rainfall\_data()*  *rainfall\_calc.cal\_avg\_ranfall(monthly\_data)* |

**Part1 Code execution:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Part2 Source code:**

**Problem statement:** The CSU Global Bookstore has a book club that awards points to its students based on the number of books purchased each month. The points are awarded as follows:

    If a customer purchases 0 books, they earn 0 points.

    If a customer purchases 2 books, they earn 5 points.

    If a customer purchases 4 books, they earn 15 points.

    If a customer purchases 6 books, they earn 30 points.

    If a customer purchases 8 or more books, they earn 60 points.

Write a program that asks the user to enter the number of books that they have purchased this month and then display the number of points awarded.

**Code:**

|  |
| --- |
| class Bookstore:      award\_points = {0: 0, 2: 5, 4: 15, 6: 30, 8: 60}        def \_\_init\_\_(self):          pass        def points\_awarded(self, numbooks: int):          sorted\_award\_points = dict(sorted(self.award\_points.items()))            awarded\_pints = 0          for key in sorted\_award\_points.keys():              if key <= numbooks:                  awarded\_pints = sorted\_award\_points[key]              else:                  print(f"Number of points awarded: {awarded\_pints}")                  return    if \_\_name\_\_ == "\_\_main\_\_":      bookstore = Bookstore()      num\_books = int(input("Please Enter the Number of books student purchased: "))        bookstore.points\_awarded(numbooks=num\_books) |

**Part2 Code execution:**

**A screenshot of a computer program

AI-generated content may be incorrect.**