

Functions Challenge 35: Loan Calculator App

Description:

You are responsible for writing a program that gathers information about a loan such as starting principal, interest rate, and desired monthly payment. From this information, your program will first determine if it is possible to pay off the loan based on the desired monthly payment. If it is possible, your program will simulate making monthly payments until the loan is completely paid off. Your program will then display statistics such as how long it took to pay the loan off, the total amount spent on the loan, and the amount spent on interest. For simplicity, we will assume that the loan compounds once a month, or twelve times in a year. Upon completion of paying off the Loan, your program will make a graph showing the rate of change of the principal to time using the matplotlib library.

Step By Step Guide:

Defining your functions:

- Define a function `get_loan_info()` which takes zero parameters.
 - Create a blank dictionary called `loan`.
 - Get user input for a loan amount.
 - Store this response, as a float, as the value for the key 'principal' in the dictionary `loan`.
 - Get user input for an interest rate.
 - Store this response, as a float, as the value for the key 'rate' in the dictionary `loan`.
 - Divide the value of rate by 100 and update the value in the dictionary to turn an example rate of 4.25% into .0425.
 - Get user input for a desired monthly payment.
 - Store this response, as a float, as the value for the key 'monthly payment' in the dictionary `loan`.
 - Set a key value pair of 'money paid': 0 in the dictionary `loan`.
 - Return the dictionary `loan`.
- Define a function `show_loan_info()` which takes two parameters, a dictionary representing the current loan, and an integer representing the current number of months that have passed during payoff.
 - Print a status for the current loan information after n number of months have passed.
 - Loop through the dictionary to print all the information in the loan.
 - There is no return value for this function.
- Define a function `collect_interest()` which takes one parameter, a dictionary representing the current loan.
 - Update the value of the principal by taking the current value of the principal and adding the $\text{principal} \times \text{rate} / 12$.
 - We are dividing by 12 because there are 12 months in a year.

- This simulates collecting interest monthly.
 - There is no need to return our dictionary since it is a mutable object and can be changed right inside of the function.
- Define a function `make_monthly_payment()` which takes one parameter, a dictionary representing the current loan.
 - Update the value of the principal by taking the current value of the principal and subtracting the monthly payment.
 - If the principal is greater than zero, add the monthly payment amount to the money paid key of the loan dictionary.
 - This implies that you were required to make a full payment to the loan for the current month.
 - If the principal is less than zero, don't add a full monthly payment but rather, add the correct amount needed to pay off the loan to the money paid key of the loan dictionary.
 - This implies that on the last month, you were not required to make a full payment.
 - Verify that your principal is zero.
 - There is no need to return our dictionary since it is a mutable object and can be changed right inside of the function.
- Define a function `summarize_loan()` which takes three parameters, a dictionary representing the current loan, an integer representing the current month number, and a float representing the initial principal of a loan.
 - Print a results summary that summarizes the following:
 - The initial loan value.
 - How many months it took to pay off.
 - How much the user spent in total rounding to two decimals.
 - How much the user spent on interest rounding to two decimals.
 - There is no return value for this function.
- Define a function `create_graph()` which takes two parameters, a data set containing month number and principal, and a loan.
 - Create a variable `x_values` and set it equal to a blank list.
 - Create a variable `y_values` and set it equal to a blank list.
 - Recall that the data set is a list. Each element of the list is a tuple (month number, principal). Therefore, for each element of the list, the month number is at index 0 and principal is at index 1.
 - Loop through the data set. For each iteration:
 - Append the current month number to `x_values`.
 - Append the current principal to `y_values`.
 - Create a graph visualization of these two variables. This is outside the scope of traditional Python. To make a graph we must import an extra library of code.
 - Type from matplotlib import pyplot

- The matplotlib is pretty big. We don't need all of it so we'll only import the part we need; pyplot.
- Create a plot using our two lists representing month number and principal.
 - `pyplot.plot(x_values, y_values)`
- Give the plot a descriptive title.
 - `pyplot.title(str(100*loan['rate']) + "% Interest" + " With $" + str(loan['monthly payment']) + " Monthly Payment")`
- Label your x axis.
 - `pyplot.xlabel("Month Number")`
- Label your y axis.
 - `pyplot.ylabel("Principal of Loan")`
- Show the created graph.
 - `pyplot.show()`

Your main code:

- Print a welcome message.
- Create a variable named `month_number` and set it equal to 0.

- Create a loan by calling the `get_loan_info()` function.
- Create a variable named `starting_principal` and set it equal to the value of the principal on the loan before you begin collecting interest or paying it off.
- Create a variable `data_to_plot` and set it equal to a blank list.

- Show the loan information by calling the `show_loan_info()` function.

- Prompt the user to begin paying off the loan.
- While the loan is greater than 0:
 - If the loan's current principal is greater than the `starting_principal`:
 - Use a break statement to end the loop.
 - You will never pay off the loan.
 - Increment the month number by 1.
 - Collect interest for the month by calling the `collect_interest()` function.
 - Make a monthly payment by calling the `make_monthly_payment()` function.
 - Append a tuple to the `data_to_plot` variable.
 - This tuple should have the current month number as its first index and the current loan principal as its second index.
 - Show the loan information by calling the `show_loan_info()` function.

- If the loan is paid off:
 - Summarize the loan information by calling the `summarize_loan()` function.
 - Create a graph showing the rate of change of the principal by calling the `create_graph()` function.
- Else:

- The loan can never be paid off, meaning, the monthly payment is less than the interest collected. You have broken out of the while loop before the loan's principal has reached zero.
 - Print a message informing the user that the loan will never be paid off.
-
- Use at least 2 comments to describe sections of your code.
 - "Chunk" your code so that is readable.
 - Use appropriate and informative variable names.
 - Format your output as below.

Example Output 1:

Welcome to the Loan Calculator App

Enter the loan amount: 40000

Enter the interest rate: 5.5

Enter the desired monthly payment amount: 175

----Loan information after 0 months----

Principal: 40000.0

Rate: 0.055

Monthly Payment: 175.0

Money Paid: 0

Press 'enter' to begin paying off your loan.

----Loan information after 0 months----

Principal: 40183.333333333336

Rate: 0.055

Monthly Payment: 175.0

Money Paid: 0

You will never pay off your loan!!!

You cannot get ahead of the interest! :-(

Example Output 2:

Welcome to the Loan Calculator App

Enter the loan amount: 10000

Enter the interest rate: 4.25

Enter the desired monthly payment amount: 250

----Loan information after 0 months----

Principal: 10000.0

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 0

Press 'enter' to begin paying off your loan.

---Loan information after 1 months---

Principal: 9785.416666666666

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 250.0

---Loan information after 2 months---

Principal: 9570.073350694443

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 500.0

---Loan information after 3 months---

Principal: 9353.967360478153

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 750.0

---Loan information after 4 months---

Principal: 9137.095994879846

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 1000.0

CUT FOR BREVITY

---Loan information after 42 months---

Principal: 301.15895867815243

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 10500.0

---Loan information after 43 months---

Principal: 52.22556332347091

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 10750.0

---Loan information after 44 months---

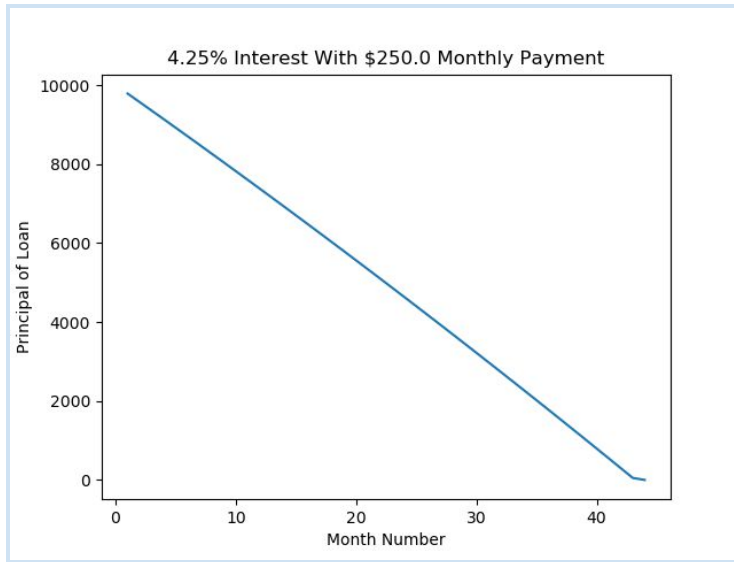
Principal: 0

Rate: 0.0425

Monthly Payment: 250.0

Money Paid: 10802.410528860242

Congratulations! You paid off your loan in 44 months!
Your initial loan was \$10000.0 at a rate of 4.25%.
Your monthly payment was \$250.0
You spent \$10802.41 total.
You spent \$802.41 on interest!



Example Output 3:

Welcome to the Loan Calculator App

Enter the loan amount: 155000

Enter the interest rate: 4.25

Enter the desired monthly payment amount: 650

----Loan information after 0 months----

Principal: 155000.0

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 0

Press 'enter' to begin paying off your loan.

----Loan information after 1 months----

Principal: 154898.95833333334

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 650.0

----Loan information after 2 months----

Principal: 154797.5588107639

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 1300.0

---Loan information after 3 months---

Principal: 154695.80016488535

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 1950.0

*** CUT FOR BREVITY ***

---Loan information after 525 months---

Principal: 979.1289480172211

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 341250.0

---Loan information after 526 months---

Principal: 332.596696374782

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 341900.0

---Loan information after 527 months---

Principal: 0

Rate: 0.0425

Monthly Payment: 650.0

Money Paid: 342233.7746430078

Congratulations! You paid off your loan in 527 months!

Your initial loan was \$155000.0 at a rate of 4.25%.

Your monthly payment was \$650.0

You spent \$342233.77 total.

You spent \$187233.77 on interest!

