

Caribbean Maritime University

Faculty of Engineering& Applied Technology

B.Eng. In Mechatronics

Computer Programming

Final Project

Names and ID Number: Vaughn Davis: 20190952

Alexander Pinnock: 20190620

Leonard Smith: 20190134

Troy Hamilton: 20190886

Group Names: Mechatronics Stream2 Year 2

**Table of Content**

1. Abstract
2. General Terms
3. Introduction
4. Designing Function
5. Psuedocode
6. FlowChart
7. Algorithm
8. Output from Test Cases
9. Conclusion & Future Scope
10. References

**Abstract**

Nowadays, people are racing to achieve a comfortable standard of living. They work hard to support their families and also to the satisfaction of his own life. It is not surprising if we see many who can afford to buy houses and school with full payment for the common interest. However not all are able to do so, those who have large families are more likely to pay in monthly installments. For example for loan calculator. Therefore our group has developed an application that will facilitate their affairs that cannot afford for the full installment for the anything. We developed an application called “Loan Calculator” to reduce their problem. The purpose of this application is to help people who want to make purchases by using monthly payments. With this application also, people will find out if they can afford school, getting a dream house etc by using this user friendly loan calculator. The user just need to insert total amount of needed, make down payments, interest rate and loan period then the application will calculate to the user how much user need to pay monthly within a certain period of time.

**Introduction**

When you are planning a purchase, but you realize that you cannot afford it from your savings or income (alternatively you are about to take an [investment](https://www.omnicalculator.com/finance/investment) decision requiring financial support), the most straightforward way to obtain the necessary money is to turn to a bank for a loan. If you are in such a situation, probably one of the first things that comes to your mind is whether you will be able to make the required payments (or in other word, installments) from your income. To solve this puzzle, you need to answer the crucial question: what will my loan payment be? With our **loan payment calculator**, **you can quickly compute the amount of money you need to devote for loan repayment over a payment period**. This device will also give you the total amount you need to pay back during the whole loan term. Alternatively, you can use it to compute the loan amount from the loan payments.

**Designing Function**

The program only contains two(2) functions:

1. amortize table () – The purpose of this function is to generate an amortization table based on the loan information that was entered by the user.

The function takes three (3) arguments:

a. amount: This is the loan amount borrowed

b. rate: The monthly interest rate

c. payment years: This is the loan duration in years

II. menu check () – This functions’ purpose is to ensure that the user is only able to enter a valid response option from the menu list that is given at the start of the program.

The function requires no arguments.

**Algorithm**

Variable initializations

The program loop is entered

The user is prompted to choose a response from the menu list provided. Valid responses are 1 through to 4.

The user’s input for the menu prompt is ran through the validation check function menu\_check() and the user only leaves this function when a valid response is received

For Option 1:

The program simply exits with a break statement

For Option 2:

If the user choose option 2 to enter some loan information into the program, that is the loan amount, the loan duration, and the interest rate

The program calculates the monthly payment, total loan amount to be repaid and the compound interest and rounds off the results to 2 decimal places

The results from the monthly payment, the total loan amount to be repaid and the compound interest are then outputted to the screen in the form of message to the user.

Then the program loop continues, and the menu options are once again displayed to the user for input

For Option 3:

The program uses the information entered by the user when option 2 was completed and passes this information to the amortize\_table(); The loan\_amount as the amount, the interest\_rate as the rate and the loan\_duration as the payment\_years.

The amortize\_table() function prints the table to the screen

Then the program loop continues and the menu options are once again displayed to the user for input

For Option 4:

The program uses the information entered by the user when option 2 was completed and passes this information to the amortize\_table(); The loan\_amount as the amount, the interest\_rate as the rate and the loan\_duration as the payment\_years.

The amortize\_table() function prints the table to the screen

Then the program loop is exited and the program is terminated

**Psuedocode**

# Declaring and nitializing variables

loan\_amount = 0.0

interest\_rate = 0.0

loan\_duration = 0.0

monthly\_payment = 0.0

total\_loan\_repaid = 0.0

compound\_interest = 0.0

number\_of\_payments = 0.0

def amortize\_table(amount, rate, payment\_years):

# Declaring and Initializing function variables

num\_of\_months = 0

total\_payments = 0.0

interest = 0.0

principal = 0.0

total\_payments = payment\_years\*12

payment = (amount\*(rate\*((1 + rate)\*\*total\_payments))) / ((1 + rate)\*\*total\_payments)-1)

table\_headers = [‘PAYMENT NO.’, ‘LOAN VALUE’, ‘INTEREST’, ‘PRINCIPAL’, ‘PAYMENT’, ‘REMAINING BALANCE’]

while amount > 0:

num\_of\_months = num\_of\_months + 1

interest = amount\*rate

principal = payment – interest

if (amount – payment) < 0:

principal = amount

table\_rows = [num\_of\_months, amount, interest, principal, payment, (amount-principal)]

amount = amount – principal

print(table)

def menu\_check():

# Declaring and initializing function variables

user\_input = 0

while True:

try:

user\_input = int(input(‘Pick a number from list’))

except ValueError:

print(‘Choose a valid response’)

continue

if user\_input <= 0 or user\_input > 4:

print(‘Valid responses only’)

continue

else:

return user\_input

while True:

#Menu System

print(“Choose a valid response from the list given below: “)

print(‘1. Exits the program’)

print(‘2. Enter loan information’)

print(‘3. View amortization table based on the entered loan information’)

print(‘4. View amortization table based on the entered loan information and automatically exits the program’)

choice = menu\_check()

if choice == 1:

break

elif choice == 2:

while True:

try:

# Gathering loan information from the user

loan\_amount = float(input(‘Enter the amount being loaned’))

interest\_rate = float(input(‘Enter the interest rate amount’))/100

interest\_rate = interest\_rate/12

loan\_duration = int(input(‘Enter the number of years to repay the loan’))

number\_of\_payments = loan\_duration\*12

except ValueError:

print(‘ERROR: Please choose a valid response’)

continue

# Calculation Section

monthly\_payment = (loan\_amount \* (interest\_rate \* (1 + interest\_rate)\*\*number\_of\_payments)) \

/ (1 + interest\_rate)\*\*number\_of\_payments) – 1))

total\_loan\_repaid = monthly\_payment\*loan\_duration\*12

compound\_interest = loan\_amount \* (1 + ((interest\_rate \* 12)/ 1)) \*\* (1 \* loan\_duration)

# Displaying the information

print(f”For the loan amount ${loan\_amount}.”)

print(f”The total amount to be repaid is ${total\_loan\_repaid}.”)

print(f”The total amount to be paid back monthly is ${monthly\_payment}.”)

print(f”The compound interest is ${compound\_interest}”)

break

elif choice == 3:

amortize\_table(loan\_amount, interest\_rate, loan\_duration)

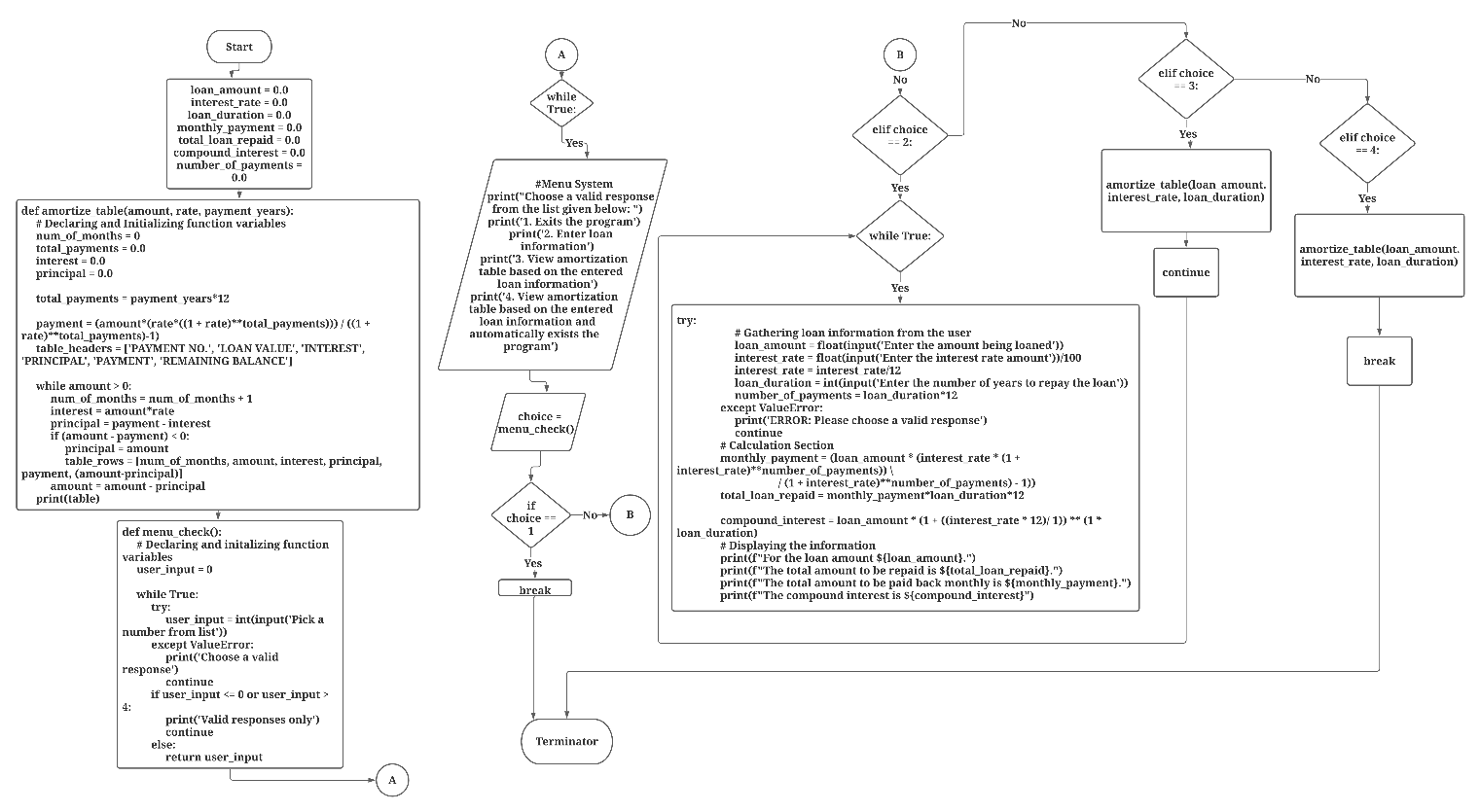
continue

elif choice == 4:

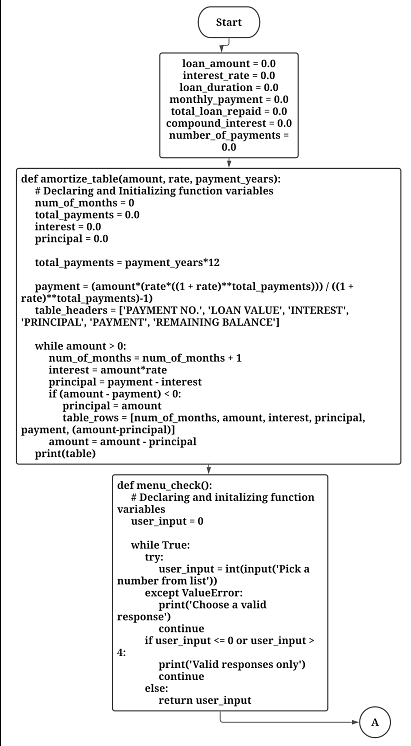
amortize\_table(loan\_amount, interest\_rate, loan\_duration)

break

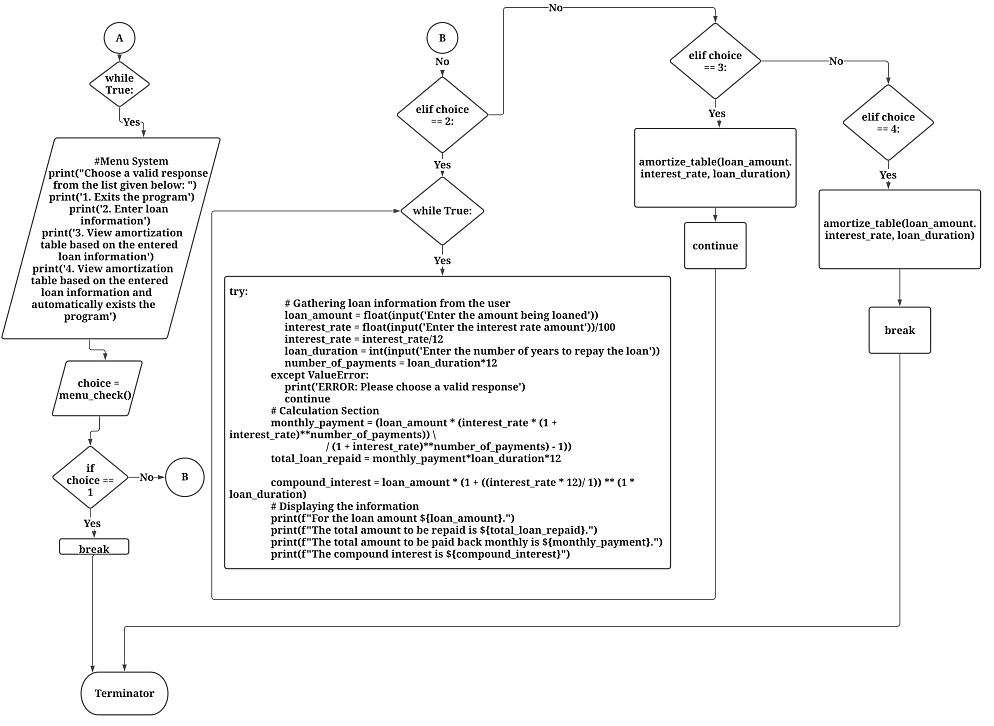
**Flow Chart**



*Figure* 1: Diagram showing overview of the entire flowchart.



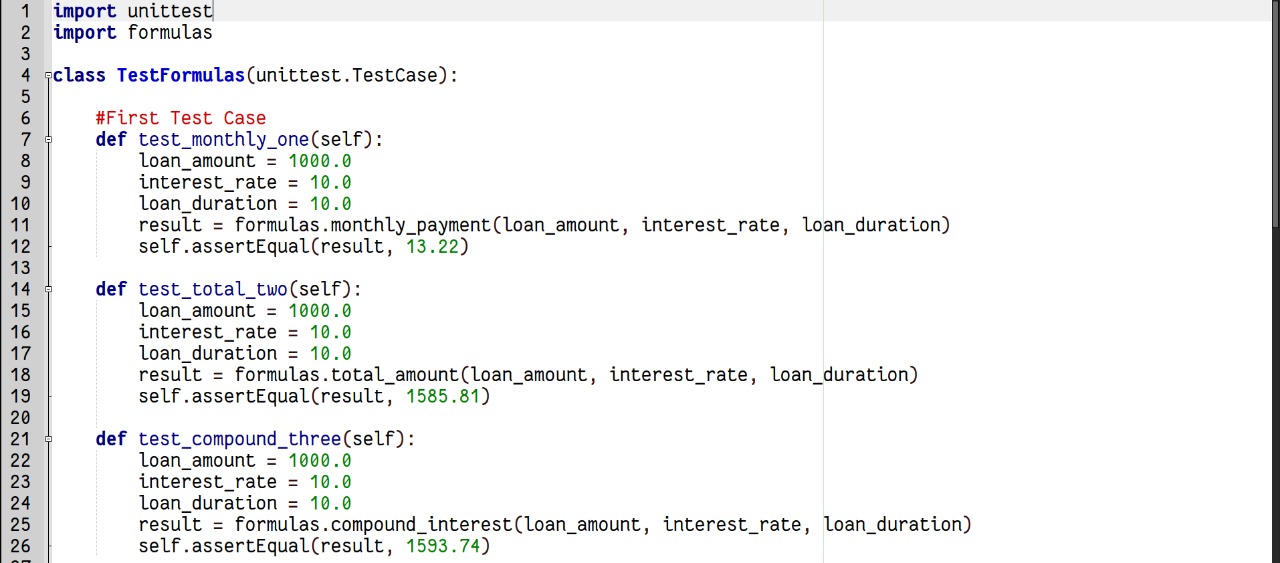
*Figure* 2: Diagram showing the first section of the flowchart (the left side of the overview).

  
*Figure* 3: Diagram showing second and third sections of the flowchart (the portion in the middle and to the right).

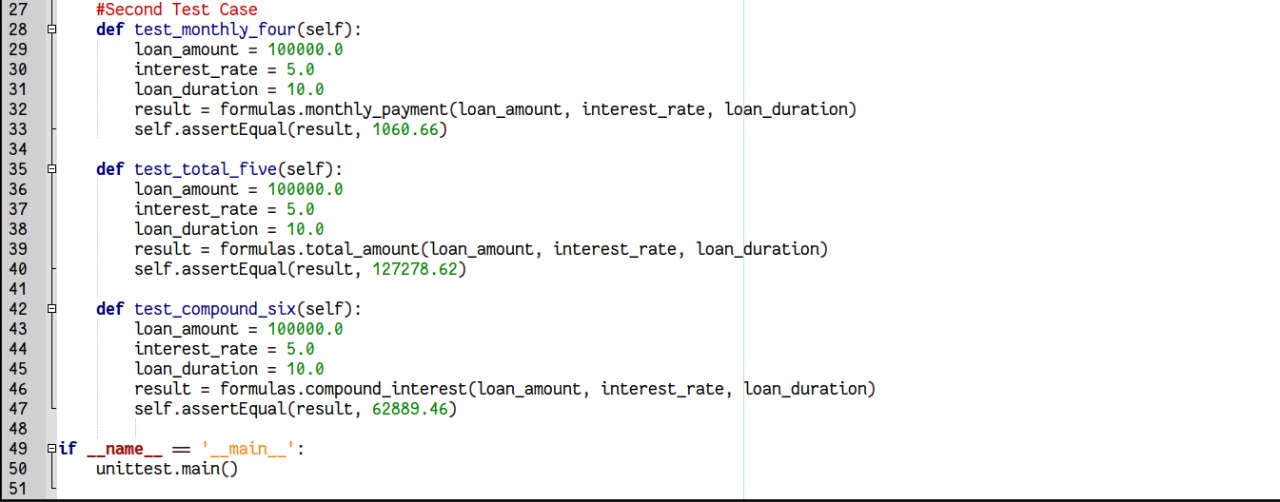
**Output From Test Cases**

Here some test cases from the Loan Calculator.

Test #1 see below:



Test #2 see below:



Testing the cases see below:



**Conclusion & Future Scope**

In conclusion to this Loan Calculator, you will be to enter information and receive calculated outputs. This task makes a helpful path for the client to increase a thought of how to perform advance computations. This project is good for the student who wants easy access to their information loan wise plus its users friendly. It is easy to understand and manipulate this project and use for education purpose only.

For the future of the program a few improvements we can add the option to allow the program to work with different compounding frequencies because right now it only works with monthly compound frequency for the monthly payments. Also allow the program to calculate compound interest based on different compounding frequencies because as of right now it only works with annually compound loans and add the option to allow the user to print out the amortization table for the previous loan information they just entered.

**Reference**

Loan Calculator In Python With Source Code. (2020, February 19). Retrieved from Source Code & Projects website: https://code-projects.org/loan-calculator-in-python-with-source-code/

‌

Python | Loan calculator using Tkinter. (2019, April 18). Retrieved January 15, 2021, from GeeksforGeeks website: https://www.geeksforgeeks.org/python-loan-calculator-using-tkinter/

‌ Smith, D. (2020, March 5). Loan Calculator App In Python With Source Code. Retrieved January 15, 2021, from ProjectNotes website: https://projectnotes.org/it-projects/loan-calculator-app-in-python-with-source-code/

‌ TheCyberBasics. (2020, August 12). Python Programming Fundamentals — Loan Payment Schedule. Retrieved January 15, 2021, from Medium website: https://medium.com/python-in-plain-english/python-programming-fundamentals-loan-payment-schedule-3e49cc1c0c86

‌ Python, R. (n.d.). How to Write Beautiful Python Code with PEP 8 – Real Python. Retrieved from realpython.com website: https://realpython.com/python-pep8/#code-layout

‌ Amortization Calculator. (n.d.). Retrieved from www.omnicalculator.com website: https://www.omnicalculator.com/finance/amortization

‌