# Financial Calculator –Documentation

## 1. Introduction

The Financial Calculator is a Python-based program designed to perform key financial calculations, including loan payment estimation, compound interest growth, and savings accumulation. This document outlines the theoretical basis, implementation details, and correct formulas used in each module of the program.

## 2. Features

The financial calculator includes the following functionalities:

- Loan Payment Calculator – Computes fixed monthly payments for a loan.

- Compound Interest Calculator – Determines the future value of an investment with periodic compounding.

- Savings Growth Calculator – Calculates the total accumulated savings with regular monthly contributions.

- Menu-driven Interface – Allows users to interactively select calculations or exit.

## 3. Theoretical Foundation

### 3.1 Loan Payment Calculation

Loan payments are typically calculated using the annuity formula for fixed-rate loans. The monthly payment (M) is determined using the following formula:

M = P × (r(1+r)^n) / ((1+r)^n - 1)

Where:

- M = Monthly payment

- P = Loan principal (amount borrowed)

- r = Monthly interest rate (annual rate / 12 / 100)

- n = Total number of months (loan term × 12)

### 3.2 Compound Interest Calculation

Compound interest follows the mathematical formula:

A = P × (1 + r/n)^(nt)

- A = Future value of investment

- P = Initial principal (starting amount)

- r = Annual interest rate (decimal form)

- n = Number of times interest is compounded per year

- t = Number of years

### 3.3 Savings Growth Calculation

When an individual makes regular monthly contributions to savings, the future savings value is computed as follows:

A = P(1 + r/n)^(nt) + PMT × ((1 + r/n)^(nt) - 1) / (r/n)

## 4. Implementation in Python

The program is structured as follows:

- A menu-driven interface allows users to choose a financial calculation.

- Three functions perform calculations: loan\_payment(), compound\_interest(), savings\_growth().

- The program loops until the user chooses to exit.

## 5. Code Implementation

import math  
  
def loan\_payment():  
 principal = float(input("Please Enter loan amount: "))  
 rate = float(input("Please Enter annual interest rate (%): "))  
 years = int(input("Please Enter loan term (years): "))  
  
 monthly\_rate = rate / 12 / 100  
 months = years \* 12  
  
 if monthly\_rate == 0:  
 payment = principal / months  
 else:  
 payment = principal \* (monthly\_rate \* (1 + monthly\_rate) \*\* months) / ((1 + monthly\_rate) \*\* months - 1)  
  
 print(f"Your estimated monthly payment is: {round(payment, 2)}")  
  
def compound\_interest():  
 principal = float(input("Please Enter initial investment: "))  
 rate = float(input("Please Enter annual interest rate (%): "))  
 years = int(input("Please Enter number of years: "))  
 compounds\_per\_year = int(input("Please Enter number of times interest is compounded per year: "))  
  
 rate\_decimal = rate / 100  
  
 amount = principal \* (1 + rate\_decimal / compounds\_per\_year) \*\* (compounds\_per\_year \* years)  
  
 print(f"Your future investment value is: {round(amount, 2)}")  
  
def savings\_growth():  
 initial = float(input("Please Enter initial savings amount: "))  
 monthly = float(input("Please Enter monthly contribution: "))  
 years = int(input("Please Enter number of years: "))  
 rate = float(input("Please Enter annual interest rate (%): "))  
  
 monthly\_rate = rate / 12 / 100  
 months = years \* 12  
  
 if monthly\_rate == 0:  
 total = initial + (monthly \* months)  
 else:  
 total = initial \* ((1 + monthly\_rate) \*\* months) + monthly \* (((1 + monthly\_rate) \*\* months - 1) / monthly\_rate)  
  
 print(f"Your total savings after {years} years is: {round(total, 2)}")  
  
while True:  
 print("\nSimple Financial Calculator")  
 print("1. Loan Payment Calculator")  
 print("2. Compound Interest Calculator")  
 print("3. Savings Growth Calculator")  
 print("4. Exit")  
  
 choice = input("Enter your choice (1-4): ")  
  
 if choice == '1':  
 loan\_payment()  
 elif choice == '2':  
 compound\_interest()  
 elif choice == '3':  
 savings\_growth()  
 elif choice == '4':  
 print("Exiting... Thank you!")  
 break  
 else:  
 print("Invalid choice! Please enter a number between 1-4.")

## 6. Conclusion

This Financial Calculator provides a practical way to compute loan payments, investment growth, and savings accumulation. The calculations are based on well-established financial formulas and incorporate compound interest and annuity principles.