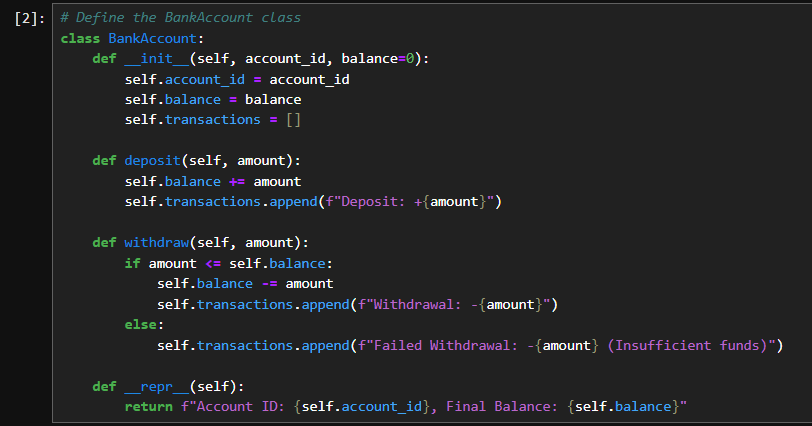
**Q:1** Generate a model in Python for representation of a bank account of type savings and balance along with transactions of deposit and withdrawals and currently create a program to generate 100 accounts with Random balance and transactions for no. of months and no. of transactions with a seed value of amount. Print all 100 accounts with the last balance and organize them by lowest to highest balance.

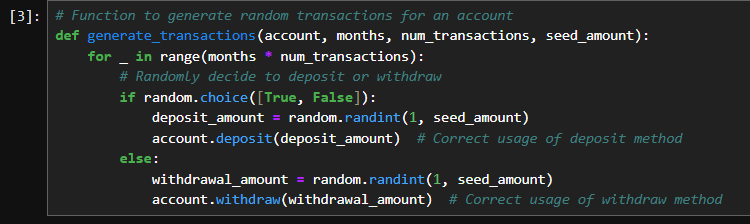
**CODE:**



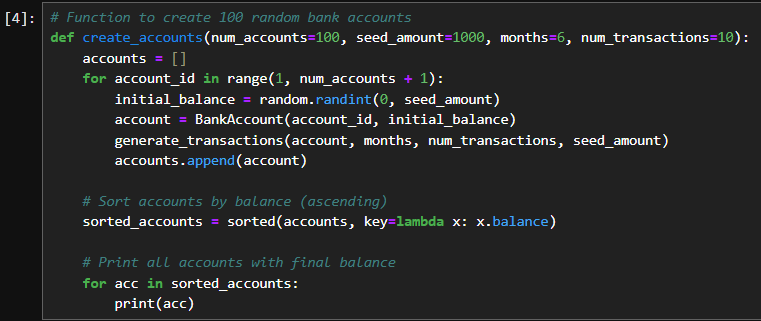
This cell imports the random module, which is used to generate random numbers throughout the program. This is necessary for creating random initial balances and transaction amounts.



This cell defines the BankAccount class with methods for handling deposits and withdrawals. It also includes a constructor for initializing account details and a \_\_repr\_\_ method for displaying account information.



This function generates random transactions (deposits or withdrawals) for a given account over a specified number of months, with a certain number of transactions per month. It uses the account's methods defined earlier to modify the account's balance.



This function creates a specified number of bank accounts with random initial balances and generates transactions for each. After processing transactions, it sorts the accounts by final balance in ascending order and prints them.

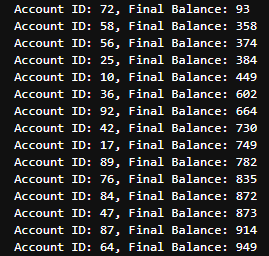


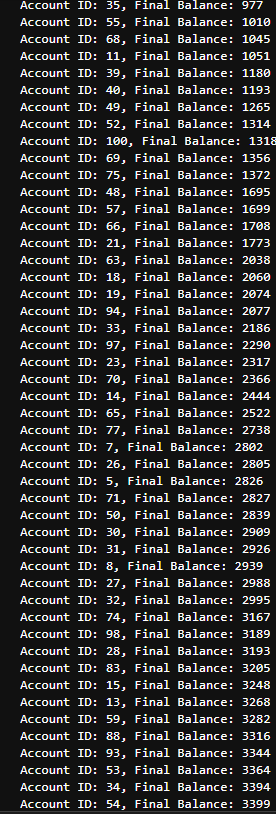
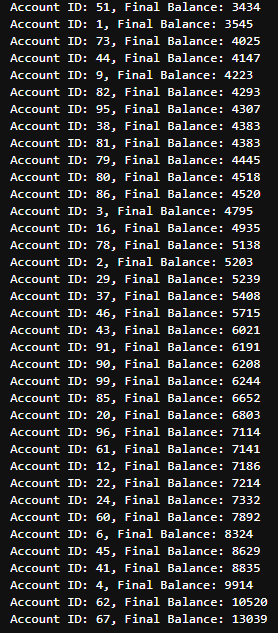
This cell sets the seed for the random number generator to ensure reproducibility of results. Using a seed value makes the random numbers predictable, which is useful for testing and debugging.



This cell calls the create\_accounts function to create 100 bank accounts with the predefined settings, process transactions, and print the sorted results.

OUTPUT:



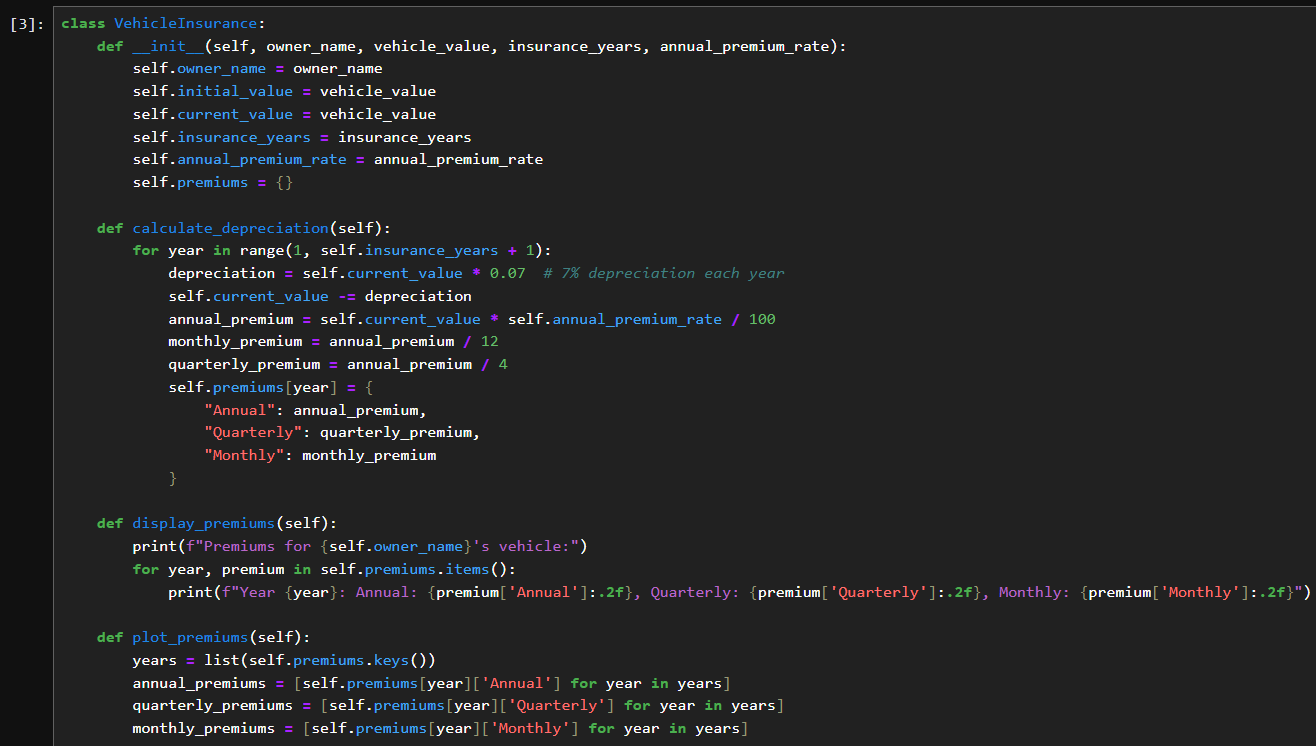
 

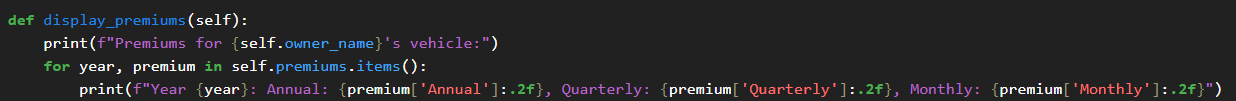
**Q:3** Generate a model for an insurance company to hold information on the insurer's vehicle, and create a chart of monthly, yearly, and quterly premiums based on no. of years of insurance where in each year, the value of the vehicle depreciates by 7%.

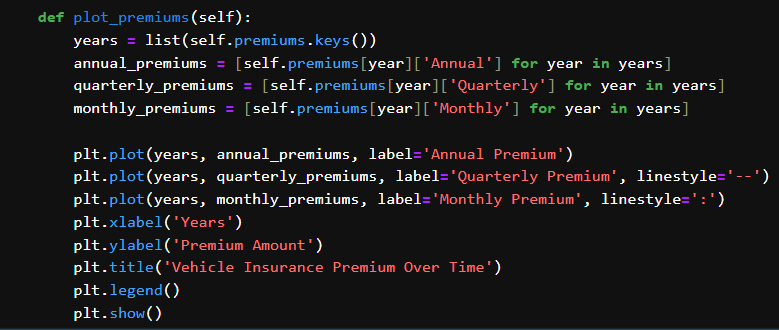
**CODE:**

****

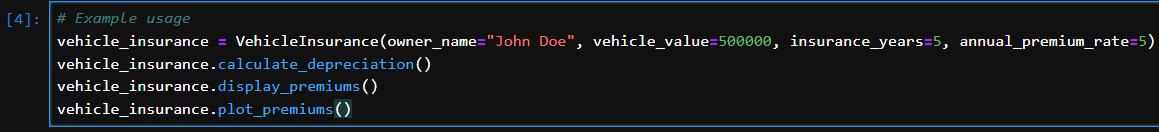
Imports the pyplot module from the matplotlib library, which provides functions for creating and customizing plots.

****

****

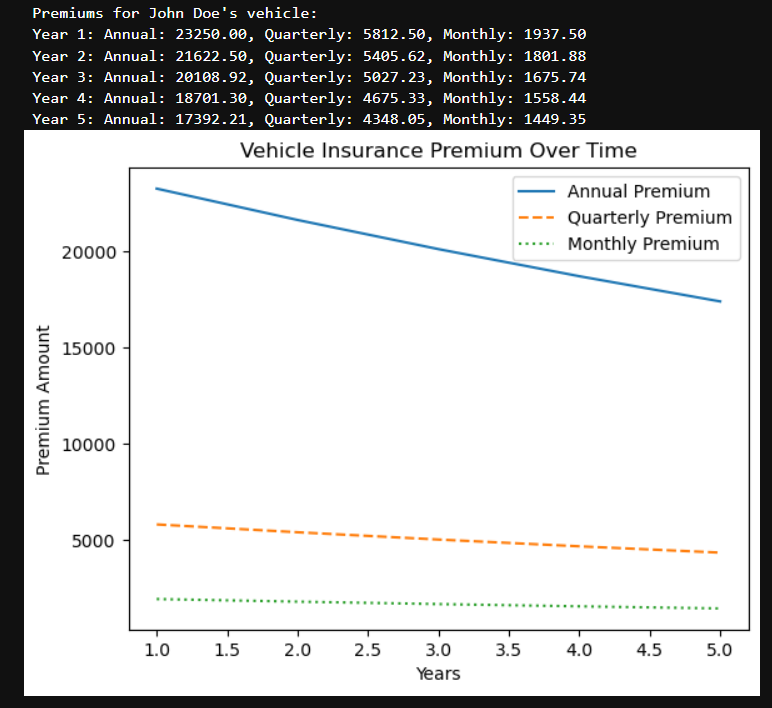
****

This cell implements the incremental development of insurance premiums for vehicle insurance through the years in the VehicleInsurance hormone. Upon creation of the class instance, parameters such as name of the owner, vehicle worth, duration of insurance and annual premium rate are usually included in the initiation of the instance. The calculate\_depreciation method includes devaluation of the vehicle at 7 percent in a year and altering the insurance premiums according to the numbers for each year. The display\_premiums method uses matplotlib to make the intervals with the enclosed annual, quarterly and monthly premiums distinct, thus allowing comparison on different types of insurance policies afforded vis-a-vis their costs through the years.



In this cell an instance of the `VehicleInsurance` class is created for "John Doe," with an initial vehicle value of $500,000, an insurance term of 5 years, and an annual premium rate of 5%. The `calculate\_depreciation` method is called to compute the premiums for each year, considering the vehicle’s depreciation. The `display\_premiums` method then prints the calculated premiums for each year, showing annual, quarterly, and monthly amounts. Finally, the `plot\_premiums` method generates a plot that visually represents how these premiums change over the 5-year period.

**OUTPUT:**

****