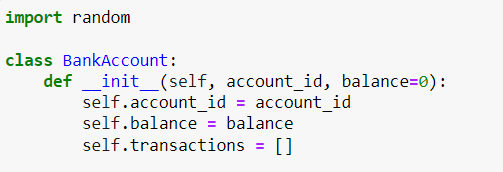
Q1. Generate a model in Python for representation of a bank account of type savings and balance along with transactions of deposit and withdrawals and currently create a program to generate 100 accounts with Random balance and transactions for no. of months and no. of transactions with a seed value of amount. Print all 100 accounts with the last balance and organize them by lowest to highest balance.

**Project 1: Bank Account Simulation with Random Transactions**

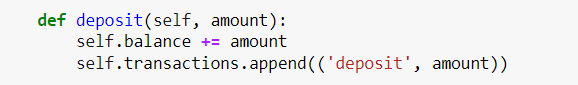
**Introduction:**

This project simulates creating and managing 100 bank accounts with random initial balances and multiple transactions (deposits and withdrawals) over several months. It also handles cases where withdrawal attempts fail due to insufficient funds. After simulating the transactions, the accounts are sorted by balance from lowest to highest.

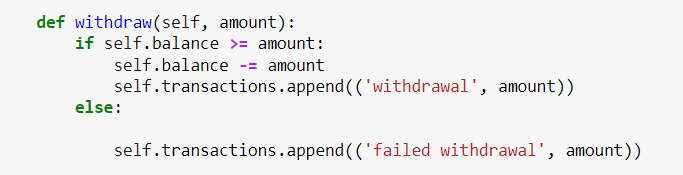


The random module is used to generate random numbers throughout the program. It is essential for creating random initial balances and simulating random transactions

This method initializes each bank account object with an account\_id, an initial balance (default is 0), and an empty list to store transactions. The account\_id is unique for each account.



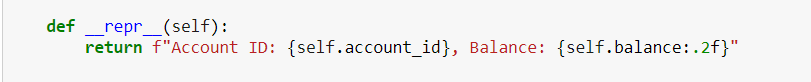
This method is used to deposit a certain amount to the account.It increases the balance by the deposited amount and records the transaction in the transactions list with the type deposit and the amount.



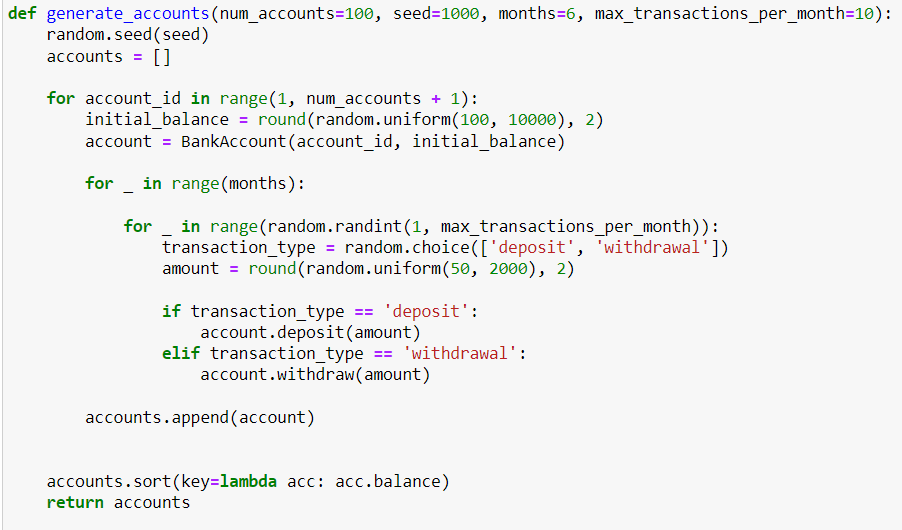
This method attempts to withdraw a specified amount from the account.

If the current balance is sufficient (balance >= amount), the withdrawal is successful, and the balance is reduced by the amount. The transaction is then recorded as a withdrawal.

If the balance is insufficient, the withdrawal fails, and the transaction is recorded as failed withdrawal.



This method defines how the account will be represented when printed. It returns the account\_id and the current balance formatted to two decimal places.



This function generates a specified number of bank accounts (num\_accounts), simulates transactions over a given period (months), and sorts the accounts by their final balance.

For each account, an initial balance is randomly generated between ₹100 and ₹10,000, rounded to two decimal places. Each account is created using the BankAccount class.

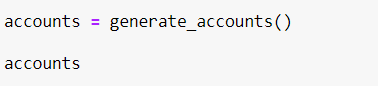
For each account, the function simulates random transactions over the specified number of months.

In each month, a random number of transactions (up to max\_transactions\_per\_month) are generated.

For each transaction, the type (deposit or withdrawal) is chosen randomly, and a random transaction amount between ₹50 and ₹2,000 is generated.

Depending on the type of transaction, either the deposit or withdraw method of the BankAccount class is called to apply the transaction to the account.

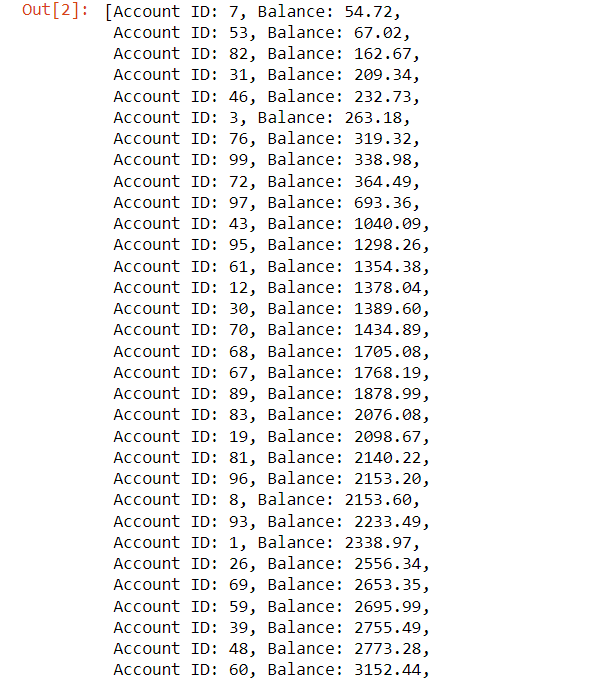
Once all transactions are simulated, the accounts are sorted in ascending order by their final balance using the sort method with a lambda function as the sorting key.

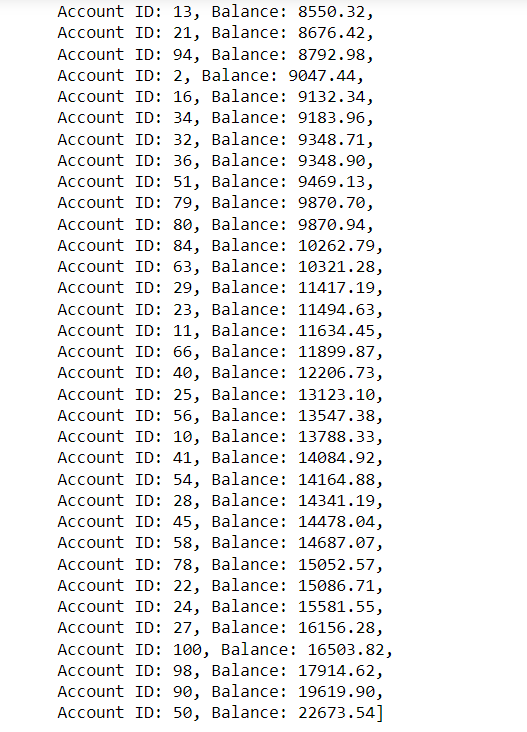
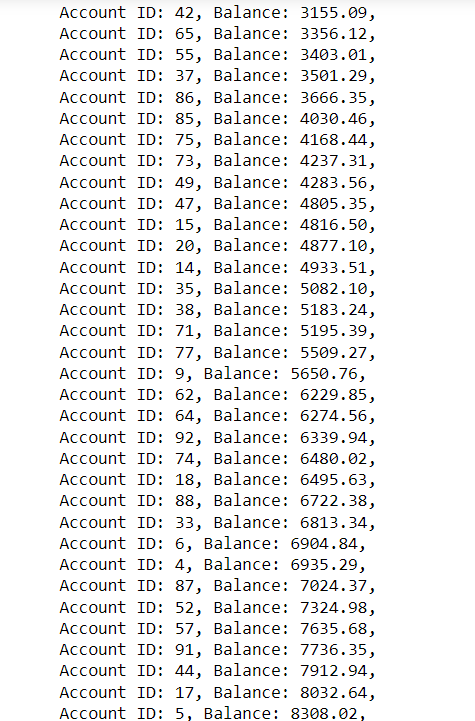


After the account generation and transaction simulation, the generate\_accounts function is called, and the list of accounts is printed, showing the account\_id and the final balance for each account, sorted from the lowest to the highest balance.

**Output:**

The code outputs the details of 100 bank accounts, each with a unique account ID and final balance. The transactions (deposits and withdrawals) are applied over several months, and the accounts are displayed in ascending order of their balance.





This Python program simulates the creation of bank accounts and the execution of random transactions. It provides a useful representation of how accounts can be handled in a banking system, including the management of deposits, withdrawals, and failed transactions due to insufficient balance. The final output provides a list of accounts organized by their remaining balance, which could help visualize account statuses at a glance.