**Assignment 1- Class and Object with Constructor in Python**

Here's another example assignment that involves creating a class with a constructor in Python. In this assignment, you'll design a simple class that represents a basic banking system. The class will have attributes such as account holder name, account number, and balance, along with methods for displaying and updating the account information.

**Task 1: Design the BankAccount Class**

Create a Python class named BankAccount that has the following attributes:

* account\_holder\_name: representing the account holder's name
* account\_number: representing the account number
* balance: representing the account balance

The class should also have the following methods:

* display\_account\_info(): displays the account holder's information
* deposit(amount): deposits the specified amount into the account
* withdraw(amount): withdraws the specified amount from the account

**Task 2: Create Objects and Test the Class**

Instantiate multiple objects of the BankAccount class and test the functionality of the class methods by displaying and updating the account information.

**Solution:**

class BankAccount:  
 def \_\_init\_\_(self,account\_holder\_name,account\_number ,balance):  
 self.account\_holder\_name = account\_holder\_name  
 self.account\_number = account\_number  
 self.balance = balance  
  
 def display\_account\_info(self):  
 print(f"The {self.account\_holder\_name} account number {self.account\_number} has balance {self.balance}")  
  
 def deposite(self,amount):  
 self.balance +=amount  
  
 def withdraw(self, amount):  
 self.balance -= amount  
  
  
Account1= BankAccount("Amit",2345678, 30000)  
Account1.display\_account\_info()  
Account1.deposite(1000)  
Account1.display\_account\_info()  
Account1.withdraw(500)  
Account1.display\_account\_info()

Results:

The Amit account number 2345678 has balance 30000

The Amit account number 2345678 has balance 31000

The Amit account number 2345678 has balance 30500