Implement a class called BankAccount that represents a bank account. The class should have private attributes for account number, account holder name, and account balance. Include methods to deposit money, withdraw money, and display the account balance. Ensure that the account balance cannot be accessed directly from outside the class. Write a program to create an instance of the BankAccount class and test the deposit and withdrawal functionality.

class BankAccount:

def \_\_init\_\_(self, account\_number, account\_holder\_name, initial\_balance=0):

self.\_\_account\_number = account\_number

self.\_\_account\_holder\_name = account\_holder\_name

self.\_\_account\_balance = initial\_balance

def deposit(self, amount):

if amount > 0:

self.\_\_account\_balance += amount

return f"${amount} deposited successfully."

else:

return "Invalid deposit amount. Please enter a positive amount."

def withdraw(self, amount):

if amount > 0 and amount <= self.\_\_account\_balance:

self.\_\_account\_balance -= amount

return f"${amount} withdrawn successfully."

else:

return "Invalid withdrawal amount or insufficient balance."

def display\_balance(self):

return f"Account Holder: {self.\_\_account\_holder\_name}\nAccount Number: {self.\_\_account\_number}\nAccount Balance: ${self.\_\_account\_balance}"

# Example usage:

if \_\_name\_\_ == "\_\_main\_\_":

# Creating a bank account

my\_account = BankAccount("123456", "Archana", 1000)

# Depositing money

print(my\_account.deposit(500)) # Output: $500 deposited successfully.

# Withdrawing money

print(my\_account.withdraw(300)) # Output: $300 withdrawn successfully.

print(my\_account.withdraw(1200)) # Output: Invalid withdrawal amount or insufficient balance.

# Displaying the account balance

print(my\_account.display\_balance())

Output:

$500 deposited successfully.

$300 withdrawn successfully.

Invalid withdrawal amount or insufficient balance.

Account Holder: Archana

Account Number: 123456

Account Balance: $200