Question:
Design a C++ class named BankAccount with the following attributes:
accountNumber (integer)
balance (double)
accountHolderName (string)
Methods:
deposit(amount): Adds the specified amount to the balance.
withdraw(amount): Deducts the specified amount from the balance, if sufficient funds are available.
displayBalance(): Displays the current balance.
Write a program that:
Creates a BankAccount object.
Prompts the user to enter account details and initial balance.
Allows the user to perform deposit and withdrawal operations.
Displays the final balance

## C++ Implementation

```
#include <iostream>
#include <string>

class BankAccount {
private:
    int accountNumber;
    double balance;
    std::string accountHolderName;

public:
    // Constructor
    BankAccount(int accNum, double initBalance, const std::string& accHolder)
        : accountNumber(accNum), balance(initBalance), accountHolderName(accHolder) {}

    // Deposit function
    void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            std::cout << "Deposit successful. New balance: $" << balance << std::endl;
        } else {
            std::cout << "Invalid deposit amount!" << std::endl;
        }
    }

    // Withdraw function
```

```
    void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            std::cout << "Withdrawal successful. New balance: $" << balance << std::endl;
        } else if (amount > balance) {
            std::cout << "Insufficient funds!" << std::endl;
        } else {
            std::cout << "Invalid withdrawal amount!" << std::endl;
        }
    }

    // Display balance
    void displayBalance() const {
        std::cout << "Account Holder: " << accountHolderName << std::endl;
        std::cout << "Account Number: " << accountNumber << std::endl;
        std::cout << "Current Balance: $" << balance << std::endl;
    }
};

int main() {
    int accNum;
    double initBalance;
    std::string accHolder;

    // Get user input for account creation
    std::cout << "Enter Account Holder Name: ";
    std::getline(std::cin, accHolder);
    std::cout << "Enter Account Number: ";
    std::cin >> accNum;
    std::cout << "Enter Initial Balance: $";
    std::cin >> initBalance;

    // Create BankAccount object
    BankAccount myAccount(accNum, initBalance, accHolder);

    int choice;
    double amount;

    do {
        std::cout << "\nBanking Menu:\n";
        std::cout << "1. Deposit\n2. Withdraw\n3. Display Balance\n4. Exit\n";
        std::cout << "Enter choice: ";
        std::cin >> choice;
```

```
        switch (choice) {
            case 1:
                std::cout << "Enter deposit amount: $";
                std::cin >> amount;
                myAccount.deposit(amount);
                break;
            case 2:
                std::cout << "Enter withdrawal amount: $";
                std::cin >> amount;
                myAccount.withdraw(amount);
                break;
            case 3:
                myAccount.displayBalance();
                break;
            case 4:
                std::cout << "Exiting program..." << std::endl;
                break;
            default:
                std::cout << "Invalid choice! Please try again." << std::endl;
        }
    } while (choice != 4);

    return 0;
}
```