Assignment: Bank Account Inheritance

Create a C++ program that models a simple bank account system using inheritance. Implement a base class `BankAccount` and two derived classes `SavingsAccount` and `CheckingAccount`.

Requirements:

1. **BankAccount Class:**
   - The base class should have private data members for the account number and account balance.
   - Include constructors to initialize the account number and balance.
   - Provide member functions to deposit and withdraw money from the account.
   - Implement a function `displayAccountInfo()` that displays the account information.
2. **SavingsAccount Class (Derived from BankAccount):**
   - Include an additional private data member for the interest rate.
   - Override the `displayAccountInfo()` function to display the account type (savings) and details including interest rate.
3. **CheckingAccount Class (Derived from BankAccount):**
   - Include an additional private data member for the transaction fee.
   - Override the `displayAccountInfo()` function to display the account type (checking) and details including transaction fee.
   -
4. Overload the `+` operator to allow adding the balances of two `BankAccount` objects. Overload the `-` operator to allow subtracting the balances of two `BankAccount` objects.

5. **Usage:**
   - In the `main()` function, create instances of `SavingsAccount` and `CheckingAccount`.
   - Demonstrate depositing and withdrawing money from both types of accounts.
   - Display account information using the `displayAccountInfo()` function for each account.

Sample Usage:

```
#include <iostream>
#include "BankAccount.h" // Include necessary header files
int main() {
    SavingsAccount savingsAccount(1001, 1000.0, 0.02);
    CheckingAccount checkingAccount(2001, 500.0, 0.5);

    savingsAccount.deposit(500.0);
    savingsAccount.withdraw(200.0);

    checkingAccount.deposit(300.0);
    checkingAccount.withdraw(100.0);

    std::cout << "Savings Account Info:" << std::endl;
    savingsAccount.displayAccountInfo();

    std::cout << "\nChecking Account Info:" << std::endl;
    checkingAccount.displayAccountInfo();

    return 0;
}
```

Submission Guidelines:

1. Implement the `BankAccount`, `SavingsAccount`, and `CheckingAccount` classes according to the provided requirements.
2. Organize your code into separate header and implementation files for each class.(bonus)
3. Test your program with various scenarios to ensure its correctness.
4. Provide a README file with explanations, usage examples, and any additional details about your implementation.