bankAccount Class

Define the class bankAccount to implement the basic properties of a bank account.   The class should store and implement the following data / methods:

|  |
| --- |
| bankAccount |
| Data* Account holders name (string)
* Account number (int)
* Account type (string, checking or savings)
* Balance (double)
* Interest rate (double, to be stored as a decimal value)
* Static member (int named anything) used to automatically assign account numbers during account creation.
 | Methods·         deposit (add money to account)·         withdraw (take money out of account)·         getInterest (balance times interest rate)·         updateBalance (adds interest amount to balance)·         print  (displays all account information)·         getAccountNumber·         getAccountHolderName·         getAccountType·         getBalance·         getInterestRate |
| Constructor:        * sets account holders name
* sets account type
* sets balance
* sets interest rate
* (account number is auto generated in constructor)
 |

Create a main program to test the class as follows:

* Declare an array of 10 components of type bankAccount.
* Prompt the user or hard code information for up to 10 customers.    For each customer, assign name, account type, balance, and interest rate.
* After all information is entered, print all account information for each customer.

Implementation Requirements

You will create three files for this assignment:

* bankAccount.h     (Class definition)
* bankAccount.cpp (Class implementation)
* bankAccountTest.cpp    (test file for your bankAccount class)

Submit source code, a screenshot with a time stamp of code execution, and a text file of the code. All code should include comments.

|  |  |
| --- | --- |
| **Grading Criteria Assignments** | **Maximum Points** |
| Program accomplishes requested operations per instructions | 40 |
| The code works and meets all assignment specifications | 30 |
| The code is organized and easy to follow and output is clear and clean | 20 |
| Uses software tools correctly and efficiently | 10 |
| **Total** | **100** |