University of Ghana

College of Basic and Applied Sciences

School of Engineering Sciences

1st Semester (2023/2024)

CPEN 201: C++ Programming

GROUP WORK

Q1.

Classes and Data Abstraction

Define the class bankAccount as an ADT to implement the basic properties of a bank account. An object of this class should store the following data:
Account holder's name (string), account number (int), account type (string, checking/saving), balance (double), and interest rate (double). (Store interest rate as a decimal number.) Add appropriate member functions to manipulate an object (they should include a function to check balance, withdraw, deposit and transfer to other accounts). Use a static member in the class to automatically assign account numbers.

Also declare an array of 10 components of type bankAccount to process up to 10 customers and write a program to illustrate how to use your class. Include a constructor and destructor in your program

Q2.

Inheritance and Composition

a. Define the class bankAccount to store a bank customer's account number and balance. Suppose that account number is of type int, and balance is of type double. Your class should, at least, provide the following operations: set the account number, retrieve the account number, retrieve the balance, deposit and withdraw money, and print account information. Add appropriate constructors.

b. Every bank offers a checking account. Derive the class checkingAccount from the class bankAccount (designed in part (a)). This class inherits members to store the account number and the balance from the base class. A customer with a checking account typically receives interest, maintains a minimum balance, and pays service charges if the balance falls below the minimum balance. Add member variables to store this additional information. In addition to the operations inherited from the base class, this class should provide the following operations: set interest rate, retrieve interest rate, set minimum balance, retrieve minimum balance, set service charges, retrieve service charges, post interest, verify if the balance is less than the minimum balance, write a check, withdraw (override the method of the base class), and print account information. Add appropriate constructors.

c. Every bank offers a savings account. Derive the class savingsAccount from the class bankAccount (designed in part (a)). This class inherits members to store the account number and

the balance from the base class. A customer with a savings account typically receives interest, makes deposits, and withdraws money. In addition to the operations inherited from the base class, this class should provide the following operations: set interest rate, retrieve interest rate, post interest, withdraw (override the method of the base class), and print account information. Add appropriate constructors.

d. Write a program to test your classes designed in parts (b) and (c).