**Mock Exam Task 2: Bank Account Management System**

**Scenario:**

You are required to implement a bank account management system. The system should handle different types of bank accounts and perform various operations such as deposits, withdrawals, and balance checks.

**Tasks:**

**Account Class:**

• Create a base class Account with the following attributes and functions:

• A std::string for the account holder’s name.

• A double for the account balance.

• A pure virtual function deposit that takes a double amount and adds it to the balance.

• A pure virtual function withdraw that takes a double amount and subtracts it from the balance, throwing an exception if the balance is insufficient.

• A virtual function getBalance that returns the current balance.

**Specific Accounts:**

• Create derived classes SavingsAccount and CheckingAccount that inherit from Account and implement the following:

• SavingsAccount allows deposits and withdrawals with no restrictions.

• CheckingAccount allows deposits and withdrawals but charges a fee for each withdrawal. If the withdrawal amount plus the fee exceeds the balance, an exception is thrown.

**Bank Class:**

• Create a Bank class with the following attributes and functions:

• A std::vector<Account\*> to hold the accounts.

• A destructor to clean up the dynamically allocated accounts.

• A function addAccount that takes an Account\* and adds it to the vector.

• A function performOperations that iterates through the vector and performs a series of operations (deposits and withdrawals) on each account, handling any potential exceptions that may arise.

**Exception Handling:**

• Implement exception handling in the performOperations function to catch and handle any exceptions thrown by the deposit and withdraw functions.