IIIT-Vadodara	
Lab 05	CSE-202 Object Oriented Programming
	Autumn'2014 [11-Sept-2014]

1. Consider creating a class Bank. An object of Bank class basically holds an unlimited number of bank account objects. A bank account object can be of type either **`SavingAccount`** or **`CheckingAccount`**, the object classes that we created in last lecture. The proposed Bank class should provide following operations.
    a. Open a new account:
    Signature: **`long open(BankAccount new_account)`**
    Parameter is a BankAccount object – it be a new SavingAccount object or a Checking account object.
    Action that happens: adds the newly created account to accounts collection of bank object.
    Returns: account number of newly opened account.
    b. Finds a bank account for given account number
    Signature: **`BankAccount find(long acc_no)`**
    Returns: returns the BankAccount object that has given account number in parameter. If no account found for given account number, then it returns null. This should return clone of the bank account in bank object.
    c. Close account for given account number
    Signature: **`BankAccount close(long acc_no)`**
    Action that happens: removes the bank account with given account number from the collection of accounts in bank object. Does not do anything if no account is found with given account no.
    Returns: returns the BankAccount that was deleted, or null if the account object is not found for given account number.
    d. You may not have any constructor for Bank class.

2. Create a BankTester class that opens about 10 different types of accounts, performs search and close operations for some of account numbers. It also performs withdraw, deposit, payinterest, deduct charges, and getBalance, equals, clone, toString operations on some of accounts in the bank object.