Class Members & Array List

The key topics for this week are writing classes with static attributes and an introduction to the ArrayList class.

TASK

1. Learn and understand the differences between the static and instance members.
2. Practice using ArrayList.

Part A.

Analyze the `BankAccount` class as below and answer the questions:

```
/**
   A bank account has a balance that can be changed by deposits and withdrawals.
*/
public class BankAccount
{
    private int accountNumber;
    private double balance;

   /**
      Constructs a bank account with a zero balance
      @param anAccountNumber the account number for this account
   */
   public BankAccount(int anAccountNumber)
   {
      accountNumber = anAccountNumber;
      balance = 0;
   }
   /**
      Constructs a bank account with a given balance
      @param anAccountNumber the account number for this account
      @param initialBalance the initial balance
   */
   public BankAccount(int anAccountNumber, double initialBalance)
   {
      accountNumber = anAccountNumber;
      balance = initialBalance;
   }
   /**
      Gets the account number of this bank account.
      @return the account number
   */
   public int getAccountNumber()
   {
      return accountNumber;
   }
   /**
      Deposits money into the bank account.
      @param amount the amount to deposit
   */
   public void deposit(double amount)
   {
      double newBalance = balance + amount;
      balance = newBalance;
   }
   /**
      Withdraws money from the bank account.
      @param amount the amount to withdraw
   */
   public void withdraw(double amount)
   {
      double newBalance = balance - amount;
      balance = newBalance;
   }
   /**
      Gets the current balance of the bank account.
      @return the current balance
   */
   public double getBalance()
   {
      return balance;
   }
}
```

a. What is the value of `b.balance` after the following operations?

```
BankAccount b = new BankAccount(1000);
b.deposit(5000);
b.withdraw(b.getBalance() /2);
```

b. If `b1` and `b2` are two objects of the `BankAccount` class, consider the following instructions:

```
b1.deposit(b2.getBalance());
b2.deposit(b1.getBalance());
```

Are the balances of b1 and b2 now identical? Explain your answer.

c. Create an ArrayList that will contain `BankAccount` objects and referred by a reference variable called `accounts.` Then add to the list three new `BankAccount` objects with account numbers of `1, 12,` and `703`, and with initial balances of 100, 200.88, and 197.23, respectively.

d. Write a testing class with `main()` method to retrieve the accounts in the list and print the details of the account with smallest amount of balance.

Part B.

Exercise 1 BankAccount class and Bank class

a. Revise the BankAccount class to include the field of “owner name” and update/define the corresponding constructors and methods
b. Define a new Bank class that consists of a list of bank accounts. This Bank class includes the methods for:
   (1) Adding a new bank account
   (2) Removal of a bank account according to the given accountNumber
   (3) Retrieval of a bank account based on a given owner name

c. Define a testing program with main() method to test your Bank class

Exercise 2 Static attributes

Write the code for a class named SportsVenuePass. This class will be used to instantiate objects that represent the right to access a performance in the sports venue. The class will have 2 private instance variables: a boolean variable named *used* and an integer variable named *passId*. The boolean variable *used* will be initialised to false and changed to true by the method : useThisPass which will be called when the pass is presented at the sports ground. When the pass is created, a unique new id number is allocated to *passId*. This id number will be generated by adding one to the previously used *passId* number, which is kept stored in a variable shared by all SportsVenuePass objects.

Exercise 3 Develop a Data Management Class

a) Write a class to represent a car used as a taxi. The attributes are: the make/manufacturer, the model, the registration licence plate, total kilometres driven, and a flag indicating whether it is currently being driven out on the streets. Include accessors, a mutator to change the flag, and a method to increase the number of kilometres driven.

b) Write a class, named CarsDatabase, which will be used to provide the responsibility of data management of a set of TaxiCar objects. Internally, it should use an ArrayList of TaxiCar. It should provide:
   i. a mechanism to add a new car into the CarsDatabase class,
   ii. a way to report the total number of cars that are being managed,
   iii. a safe way to obtain an ArrayList of all the cars that are being managed,
   iv. a safe way to obtain an ArrayList of all the cars that are currently out on the streets
   v. a safe way to obtain an ArrayList of all the cars that are currently available to be used by taxi drivers (i.e. that are not out on the streets)
   vi. a safe way to obtain an ArrayList of all the cars of a particular specified make/manufacturer.
   vii. A safe way to obtain an ArrayList of all the cars that have driven at least as many kilometres as some specified value.

c) Write a main-program class which:
   i. Creates an instance of the CarsDatabase.
   ii. Creates several different cars (maybe 7 or 8), and adds these to the CarsDatabase. Before adding them, modify a few to make them seem to be ‘in-use’, and set the distances travelled for some of them to be higher than the default distance of 0.
   iii. Test that each of the methods of the CarsDatabase class is working correctly