CST8132 Object Oriented Programming

Lab 2 Association of Objects - Composition

Objectives:

In this lab, we will be creating the basis of a Librarian Management system. We will be creating a few classes and will be building upon them in future labs. The resulting Java Program will implement association between classes - a relationship called composition used to support code re-use.

Defining a class, instance variables, constructors, setters and getters

There will be three classes in this lab: Librarian, Library, and TestComposition.

Classes

Part I: Librarian Class:

Instance variables

- ***librarianID***, of type **int**, initialized to **0**.
- ***firstName***, of type **String**, initialized to **empty**.
- ***lastName***, of type **String**, initialized to **empty**.
- ***serviceSection***, of type **String**, initialized to **empty**
- ***salary***, of type **double**, initialized to **0.0**;

Note that an empty string is effectively just “”.

Class constructor:

Five parameters, in the order as listed in the Instance variable section. Those five variables should be initialized by the values passed in the constructor.

Setters/Mutators

- ***setServiceSection***, with one parameter. Set the value of ***serviceSection*** to the value that was passed.

Getters/Accessors

- ***getFirstName***, with no parameters, that returns the value of ***firstName***.
- ***getLastName***, with no parameters, that returns the value of ***lastName***.
- ***getServiceSection***, with no parameters, that returns the value of ***serviceSection***.
- ***getLibrarianID***, with no parameters, that returns the value of ***librarianID***.
- ***getSalary***, with no parameters, that returns the value of ***salary***.

Implementing composition form of association between Library object and Librarian object

The Library can have many librarians and has many service sections such as ‘media resources’, ‘kids place’, ‘study area’ etc. A librarian works only in one service section. A service section can have many librarians.

Assumption to support composition: A librarian cannot exist without the Library.

Write the definition of class Library:

Library Class

Instance variables

- myLibrarians, an array of type Librarian.

Class Constructor

One parameter, of type integer, which will define the size of the myLibrarians array.

Methods

- ***getListOfLibrarianDetails***, no parameters. Returns the array ***myLibrarians***.
- ***addLibrarian,*** with parameters of type Librarian and integer. This method adds the Librarian object to the ***myLibrarians*** array at the specified location in the array.

Part III – TestComposition Class

Methods

Main only. In it, you must create four Librarian objects:

222 Becky Linpay Media 1500.00	333 Michael Steve Kids place 1400.00	444 Jose Lanman returns 1200.00	555 Your lname Your fname Study area 1800.00

Add these four Librarian objects to a new Library object, which you have correctly initialized with the correct constructor.

Retrieve the array of Librarian objects from your Library, using the ***getListOfLibrarianDetails*** method.

In a for loop, iterate through the array, accessing each individual Librarian, and printing out the details to the console. Your output should look more-or-less like this:

LibrarianID	**LastName**	**FirstName**	**ServiceSection**	**Salary**
=========	=========	=========	==========	======
222	Becky	Linpay	Media	1500.00
333	Michael	Steve	Kids place	1400.00
444	Jose	Lanman	returns	1200.00
555	**Your lname**	**Your fname**	Study area	1800.00

The Lab Professor will examine your code with an eye to seeing how closely you followed this document. Matching the output format is of least importance.

Bonus:

You may add one field to the Library class. Ensure that no invalid data can be sent to Library to cause it to crash. (Hint: There are two sources, both of them having to do with the array.)