

48024 Object Oriented Design

Self-Study Module: Object Design

This module examines how to design “good” objects and classes by investigating the design principles of low coupling and high cohesion.

Learning objectives:

- To develop skill in understanding and applying design principles of low coupling and high cohesion

References:

- Chapter 10, Simon Bennett, Steve McRobb, Ray Farmer, "Object-Oriented Systems Analysis and Design using UML", McGraw-Hill, 2nd edition, 2002.
- Chapter 17, Craig Larman, “Applying UML and Patterns”, Prentice-Hall, 3rd edition, 2005.

Source Code:

- AdventureGame BlueJ Project (Downloadable from UTSONline OOD folder Course Documents/Self-Study Modules)

Contents:

Exercise 1 - Interaction coupling and class/operation cohesion	2
Exercise 2 – Class cohesion.....	5
Exercise 3 - Laboratory.....	6

Exercise 1 - Interaction coupling and class/operation cohesion

In this exercise, you will explore interaction coupling and class/operation cohesion through the examination of responsibilities. We will use an example from Bennett where there is a requirement to find out if an employee works on a campaign with a particular title. This is illustrated in the class diagram (Figure 1) by the association `CreativeStaff workOnCampaign Campaign`.

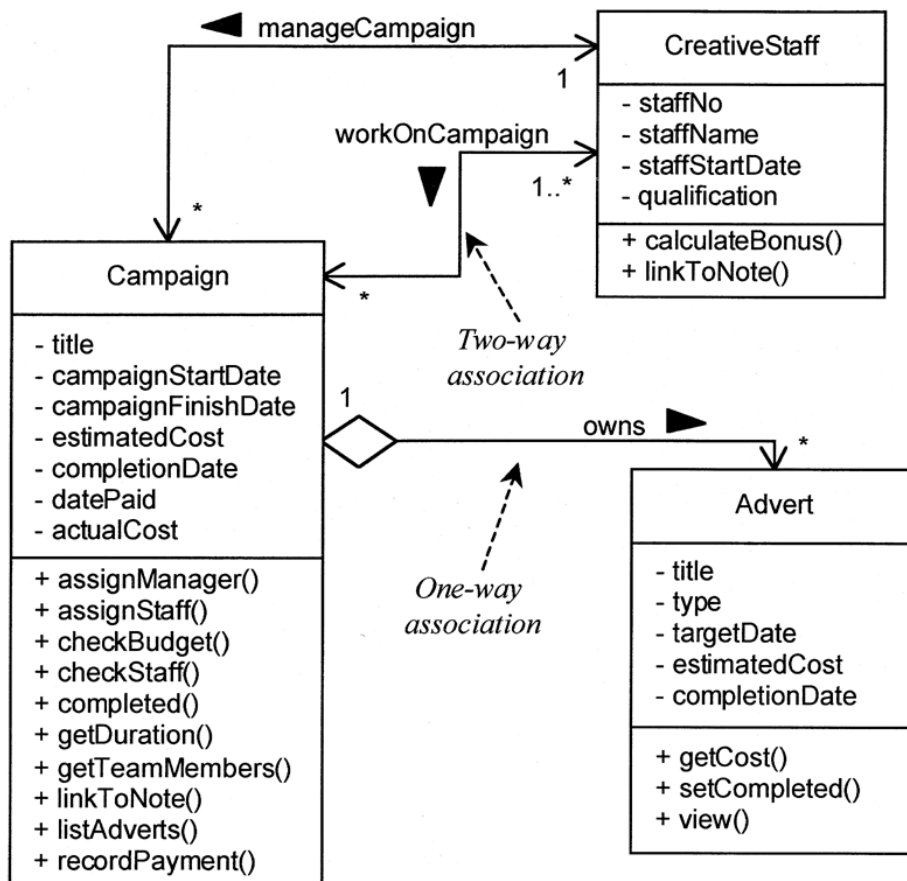


Figure 1. Class diagram (Ref: Figure 14.12 Bennett, p.358)

We will now examine one direction of this association, namely the *one-way* association between the `CreativeStaff` and `Campaign` classes. This is illustrated in the following class diagram fragment, Figure 2. It includes a collection class, `CampaignCollection` which contains object identifiers for the set of campaigns worked on by a creative staff member. As suggested on p.361 Bennett, there are at least two alternative strategies for searching a collection class. We will explore coupling and cohesion for these two alternative search strategies.

48024 Object Oriented Design
Self-Study Module: Object Design

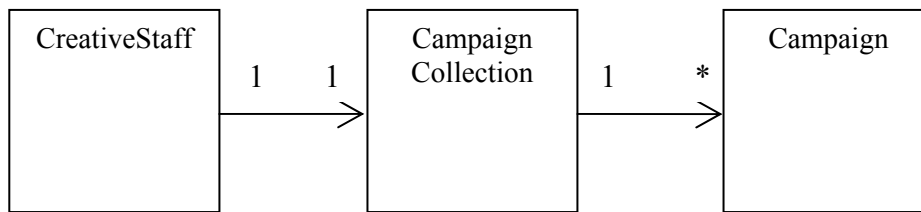
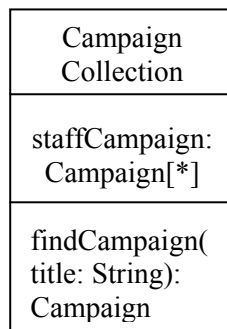


Figure 2. Class diagram fragment

For the strategies below, write down the *responsibility* of each class – this gives an indication of the operation and class cohesion. Identify the *type* and *number of messages* sent by each class – this gives an indication of the coupling between classes.

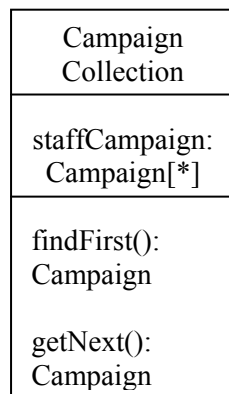
Strategy A

The collection class, CampaignCollection provides an operation, findCampaign() that is responsible for finding the Campaign object with matching title. This strategy hides the iteration.



Strategy B

The collection class, CampaignCollection provides two operations, findFirst() and getNext() that allow the user to iterate through the collection, returning the next Campaign object with each successive call to getNext().



Discussion ...

Which strategy provides higher cohesion?

Which strategy provides lower coupling ?

What are the trade-offs between the two strategies? Consider in terms of reuse and extensibility.

Exercise 2 – Class cohesion

In this exercise, you will explore coupling and cohesion by applying Larman's principles for assignment of responsibilities. For your design model from the Assignment, identify the responsibilities of each class and indicate the level of coupling and cohesion for each class. Select one of the principles for assignment of responsibilities (Table 1) to redesign the class to achieve higher cohesion and/or lower coupling. Otherwise present an argument for why the class should remain as it is.

Table 1 - Principles for assignment of responsibilities

Principle	Creator
Problem	Who creates an A?
Solution	Assign class B the responsibility to create an instance of class A if one of these is true (the more the better): B `contains' A B records A B closely uses A B has the initialising data for A
Principle	Information Expert
Problem	What is a basic principle by which to assign responsibilities to objects?
Solution	Assign a responsibility to the class that has the information needed to fulfil it.
Principle	Low Coupling
Problem	How to reduce the impact of change?
Solution	Assign responsibilities so that (unnecessary) coupling remains low.
Principle	Controller
Problem	What first object beyond the UI layer receives and coordinates ('controls') a system operation?
Solution	Assign the responsibility to an object representing one of these choices: represents the overall `system' or a major subsystem, e.g. Bank represents a device that the software is running within, e.g. BankCashMachine represents a use case or session, e.g. TransactionHandler
Principle	High Cohesion
Problem	How to keep objects focused, understandable and manageable, and as a side effect, support Low Coupling?
Solution	Assign responsibilities so that cohesion remains high.

Exercise 3 - Laboratory

Learning objectives:

- To gain skill in writing code in Java that guarantees low coupling and high cohesion of classes
- To identify simple good design and development guidelines that facilitate easier maintenance and extensibility of Java applications.

Source Code:

- AdventureGame BlueJ Project (Downloadable from UTSONline OOD folder Course Documents/Self-Study Modules)

In this laboratory exercise you are required to modify the code of the AdventureGame¹ project to lower the level of coupling between classes and increase the level of cohesion.

Task 1. Understanding AdventureGame

The AdventureGame project is a simple framework for an adventure game. In this version, it has a few rooms and the ability for a player to walk between these rooms.

Open the project, compile it, create an object of the class Game and call the play method of it to start playing. After that, look at each one of the classes to understand what their attributes and methods are for. Read the comments since they help you understand the code.

Task 2. Evaluating and decreasing level of coupling

In this version of the project there are links between classes that makes them more or less dependant on each other. Examine the code of every class, identify cases of coupling with other classes and determine the nature of a link. Then try to figure out if there is a way of eliminating or changing the definition a link that increasing the level of coupling without affecting the functionality of the system.

Some code issues you can look at are:

Use of encapsulation – Do the classes follow the principle of encapsulation? If not, what needs to be done in other to apply it correctly? Why does this affect the level of coupling? Make changes to the code in order to apply encapsulation where needed.

Assignment of responsibilities – Look at the methods of a class and evaluate whether the class should be responsible for carrying out that particular task. Remember that one of the principles of well-defined classes is to allocate a responsibility to the class that stores the data to be manipulated or accessed. If you take this into account you can reduce the level of coupling since a class would not depend on the information of another class to perform a task. (Note: not all the links are meant to be broken, some of them are needed in order to keep good cohesion)

Duplicated code – If two new directions (e.g. *up* and *down*) are to be added as new possible exits of a room, what changes and in which classes need to be made? Why? Is there a way of not repeating so much code? Maybe by changing how data is stored

¹ This project is based in the World of Zuul game know as ‘Colossal Cave Adventure’ as well.

48024 Object Oriented Design

Self-Study Module: Object Design

and structured? Analyse this specific example and then make the needed changes to the code. Review the rest of the code for more cases of code duplication.

Task 2. Evaluating and increasing level of cohesion

Cohesion can be evaluated at two levels:

Cohesion of methods – methods should perform one single well-defined task. Are there any methods in the project that do more than one task? If so, what can you do to fix it?

Cohesion of classes – cohesive classes represent one well-defined entity. Is this the case for all the classes in the project? What would you do if you have to add an extra functionality to a class and maintaining high cohesion at the same time?

Document all the modifications you did, explaining the reasons and effects of each one of them. Answer all the questions posed in this exercise.