

Object Orientation with Design Patterns

Module Code:

Module Title:	Object Orientation with Design Patterns
Assignment Type:	Continuous Assessment (Group – Max. 3, Min. 2)
Project Title:	Commercial Trading
Project Date:	14 th March 2018
Assignment Compiler:	Mr. Mark Morrissey
Weighting:	50% multiple submissions : 5% / 15% / 30%
Due Date:	27 th April 2018 at 11:59pm (Moodle Assignment Link)

Module learning outcome:

1. Assess application design requirements and improve existing application designs by applying a custom implementation of object orientated design patterns (Linked to PLO 5)
2. Determine when design patterns should not be used during the development process and how they can hinder larger design plans (Linked to PLO 1, 2)
3. Justify the differences between the design patterns that are used in industry and how they are altered for different applications (Linked to PLO 8)

Module Objectives

1. How to improve application code reusability by designing modular object oriented code.
 2. The design and implementation of design patterns in the code, allowing the student to design structured applications using predefined solutions to recurring application design problems.
 3. How to create modular, easy to maintain code by using design patterns during the design process of applications, allowing the student to create reusable modular code.
 4. The process of implementing behavioural and structural design patterns, allowing the behaviour of applications to be specifically defined, and the control in the application to be predictable and fault tolerant.
 5. The problem solving skills needed to assess the relevance of specific pre-defined design solutions to transfer conceptual ideas into well-defined and designed solutions.
-

Project Brief: Commercial trading

Design and implement a program to allow sales of products between three companies.

BIGa produces : A / BIGb produces : B / BIGc produces : C

Each of the companies produce one product, each company must trade with the other two to buy products.

Company BIGa is a supplier of A's. It buys B's and C's

BIGa has 100 depots each containing a varying number of A's (min stock of A's = 15)(max stock 50)

Each depot can hold a minimum of 3 B's and C's and a maximum of 40

Company BIGb is a supplier of B's. It buys A's and C's

BIGb has 100 depots each containing a varying number of B's (min stock of B's = 15)(max stock 50)

Each depot can hold a minimum of 3 A's and C's and a maximum of 40

Company BIGc is a supplier of C's. It buys A's and B's

BIGc has 100 depots each containing a varying number of C's (min stock of C's = 15)(max stock 50)

Each depot can hold a minimum of 3 A's and B's and a maximum of 40

Functional parameters:

A user must be able to select which company to trade as

A user must be able to select to allow the program to trade autonomously

A depot cannot go below its minimum stock of its native product

A depot cannot store above its maximum stock of its native product

A depot cannot go below its minimum stock of its purchase products

A depot cannot store above its maximum stock of its purchase products

All depots must attempt to trade with every applicable depot (every 'A' depot must attempt to trade with every 'B' and 'C' depot etc.)

Product prices and delivery prices are random numbers between 1 & 10

Every depot has a cash allowance minimum 50 maximum 100

Required output of program:

1. Detailed information on each user depot that traded for each trade completed:
 - Depot buying products
 - Depot they are buying from
 - Cost of products
 - Cost of delivery
 - Total cost of doing business
2. Detailed information on user company that traded:
 - Total trade for each company
 - Total amount of each type of products purchased
 - Total cost of purchase price for products
 - Total cost of delivery for products
3. Profit and loss for each company (user selected and autonomous)
 - Company that spent the most and company that made the most

You are required to implement the preceding program specification, in order to do this you must implement / customise at least three Object Oriented Design Patterns. Use best practice in Object Oriented software development and demonstrate knowledge of Object Oriented programming principles.

Notes : **(program must run, no debugging will be attempted)**

Group size : 3 members maximum

2 members minimum

This is a group allocation of marks, no marks allocated for individual roles

Along with your reports / documentation you must only submit a **netbeans project folder**; all submitted materials should be compressed into a folder named after a student name / number from one of your group: eg firstName lastName 1234567.zip. Only one submission per group.

Submissions:

21st March: 5%

Requirements gathering and analysis: detailed use case analysis (user walk through), functional analysis and preliminary class diagrams.

4th April: 15%

Design document: To include details of the design patterns to be used, final functional analysis, class diagram.

27th April: 30%

Working program, project report detailing design choices and implementation of same. Report from each group member detailing their roles within the project. API documentation.

Marking Scheme Summary

Description	Weighting
Requirements gathering and analysis	5%
Design document	15%
Final program (including documentation)	30%
TOTAL	50%