Sept 2019 (B)

Describe the following technologies and their purpose.

XML Path Language (XPath)

- A declarative language introduced by W3C to access and navigate the content of the tree structure
- Its purpose is to define part of an XML document and perform simple calculations
- Constraint can be applied to limit the part extracted

Extensible Stylesheet language (XSL)

- It is XML vocabulary and organized into 2 languages which are XSL-FO (Extensible Stylesheet language - Formatting Objects) and XSLT (Extensible Stylesheet language Transformation)
- Its purpose is to present XML data in an easily readable format and transform the content of an XML document into a result document written in a new format

Extensible Stylesheet language Transformation (XSLT)

- It is a language that use XPath for transforming XML documents into other XML documents
- It is used for text-based output such as HTML and XHTML for creating web pages and takes an XML file and rewrite it as a new XML document with a different structure

Evaluate 4 benefits of software design patterns

- **Facilitate reuse** - Reuse the design patterns can prevent subtle issue that might cause major issue and improve the code readability for the developer who familiar with the design pattern
- **Define a shared vocabulary for discussing design** - It makes the communication among developers more efficient as they can immediately think of the high level design in their mind when they know the name of the pattern that solves a common occurling design problem during discussion.
- **Demonstrate concepts and principles of a good design** - The class created are loosely coupled and they are reusable and extensible in multiple projects
- Knowing popular design patterns **makes it easier for developers to learn the class library** that uses design patterns concepts, thus can speed up the software development time to assure high quality software being developed.

Discuss the importance of each of the following recommended software security practices

Clearly defined roles and responsibilities

- Each member in the development team clearly knows their roles and each role has its own access capabilities. In other words, it enforces least privilege in which the user is restricted to only the functionality, data and system information that is required to perform their task. This greatly reduces the attack surface by eliminating unnecessary access rights for each role level that will compromise the software security.

Provide development team with adequate software security training

- It can help to develop a security-focused culture and empower employees to follow security protocol during software development. A software security training can be used as a preventative control to secure the software. It provides training for developers such as how to prevent common vulnerabilities (i.e social engineering) proactively.

Implement a secure software development lifecycle

- It helps to discover and reduce vulnerabilities in the early stages which make software become more secure as security is a continuous concern in secure SDLC. It includes software assurance activities such as architecture analysis during design , code review during development and penetration testing before software releasing. Therefore, we can detect the software design flaws easily before they are coded into existence.

Establish secure coding standards

- It is a principle and guideline that are used to prevent security vulnerabilities in the software. With secure coding standards, it helps to mitigate the vulnerabilities and risks associated with software product development process such as security breach

Build reusable object libraries

- Reusable object libraries ensure the standard compliance accomplished with prior software development processes. To illustrate, it can increase the overall quality of a software system. Reusable object libraries are often used multiple times and are therefore proven to provide better code structure quality compared to newly developed libraries.

Verify the effectiveness of security control

- Security controls are safeguard or countermeasures that can help in reducing or mitigating the risk associated during software development process. Once controls have been implemented, the organization needs to monitor its control environment to confirm that controls remain effective to ensure it is still capable of protecting the confidentiality, integrity and availability of information.

Establish secure outsourcing development practices

- Establish secure means such as continuously communicating with the outsourcing team for the security standards on how data is handled and transferred to them, the level of access control over sensitive data so that they can put these security principles into practice to ensure software secureness during software development lifecycle. This is to protect the confidentiality of information which can avoid cyber attacks that will cause data to leak.

OCT 2021

Question 1 (b)

4 Advantages and one disadvantage of strongly typed language over weakly typed language

Advantages

Catch errors earlier

Strongly typed language can catch errors during compile time, thus making it easier to debug and test the program and more likely less bugs will occur in the program.

More manageable code

Using strongly typed language, the large code base can be more organized and manageable

Clearer code

Strongly typed language can clarify how things are used in the program. For example, the behavior of operation is more predictable by looking at the data type being assigned

More efficient code

Compiler can use the type information to make the code become more efficient

Disadvantages

Loss of some flexibility as strongly typed language has stricter typing rules at compile time compared to weakly typed language that only catch error during run time, which implies that error and exception are more likely to occur during compilation.

Most of these rules affect variable assignment, function return value, procedure argument and function calling.

Question 2

Refer to sept 2019 (B) paper for the answer. They are the same

Question 3

Consider the following classes for a weather monitoring system:

Class	Description
`WeatherData`	Stores the most recent temperature, humidity, and barometric pressure measurements.
`CurrentConditions`	Displays the current temperature and humidity.
`WeatherStatistics`	Displays the current average, maximum and minimum temperatures.
`ForecastWeather`	Displays a forecast message by comparing the current pressure with the last pressure.

The system should provide three displays, namely the current conditions, weather statistics and a weather forecast, all updated in real time as the `WeatherData` object acquires the most recent measurements. The system should also be extensible such that it allows additional weather displays to be added to the system in the future.

Propose an appropriate design pattern for the above scenario. Your answer should include

- An introduction to the proposed design pattern.
- A UML class diagram to represent how this design pattern may be implemented for this scenario. You may include additional interfaces and classes that are required for the design pattern.
- A detailed description of how the various classes in your UML class diagram will interact with each other.

(5 + 10 + 10 marks)
[Total: 25 marks]

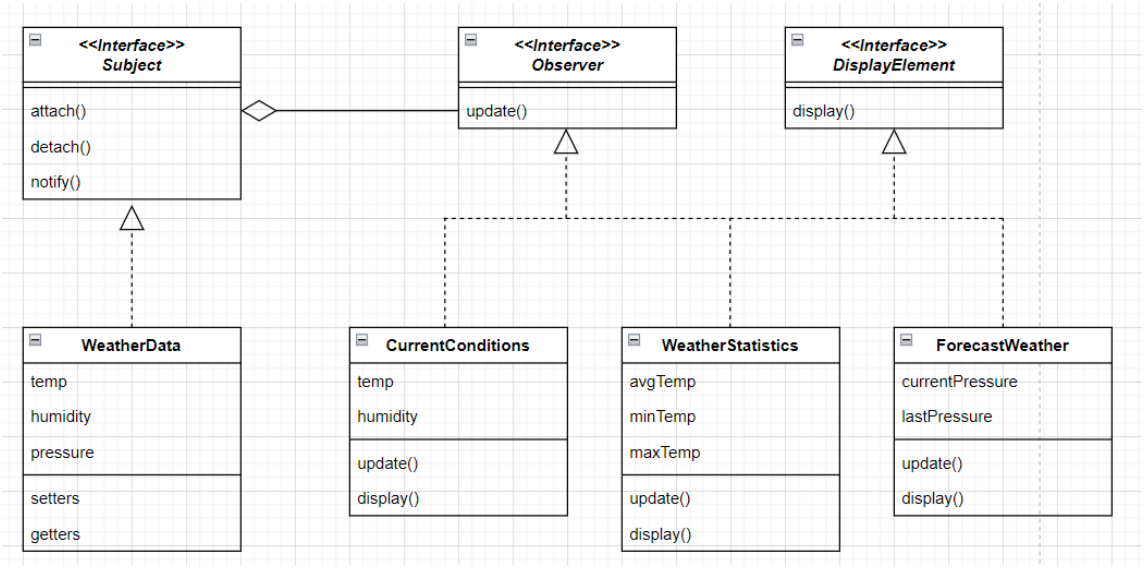
Proposed design pattern

Observer pattern

Introduction

Observer is a behavioral pattern to describe how objects interact and how responsibilities are distributed. It is a one-to-many dependence in which when a subject changes its state, the dependent objects will be notified and updated accordingly. Subject can be defined as the object that will frequently change its state and upon which other objects depend whereas observer can be defined as the object that depends on a subject and updates according to its subject state. It can also be known as publish-subscribe.

UML Class diagram



Detailed description of how the various classes in UML diagram interact with each other
Subject
The Subject interface class will know its observer. Any number of observers, in this case, CurrentConditions, WeatherStatistics and ForecastWeather observer class will observe the Subject. It provides an interface for attaching and detaching these observer objects.

ConcreteSubject: WeatherData
This class stores state of interests which are temperature, humidity and pressure to Concrete Observer classes (CurrentConditions, WeatherStatistics and ForecastWeather). It will send notification to its observer when its state changes.

Observer
The Observer interface class defines and updates references to CurrentConditions, WeatherStatistics and ForecastWeather observers that would be notified of changes in a subject.

ConcreteObserver: CurrentConditions, WeatherStatistics and ForecastWeather
All of them maintain a reference to the WeatherData object and stores state that stay consistent with it. They implement the Observer updating interface to keep its state consistent with the WeatherData to display the current temperature and humidity for CurrentConditions observer, display the current average, maximum and minimum temperatures for WeatherStatistics observer and display a forecast message by comparing the current pressure with the last pressure for ForecastWeather observer.

12 February 2019

Question 1 (b)

Implementing XML parsing using Document Object Model (DOM) parsers is the most efficient way

Do you agree with this statement? Justify your answer

- I'm not agree with this statement.
- The XML parsing implementation using which XML parser should vary based on the situation.
- To illustrate, if the user wants to manipulate the XML or access the XML many times, DOM is more efficient as it is faster once the tree is constructed as the entire XML document is already stored in the memory. Hence, the user can navigate and traverse the DOM document in any direction in a short period of time.
- Otherwise, if the XML document is large or the user only needs to extract a few elements from the document, it is more suitable to use the Simple API for XML (SAX) parser. Another suitable situation includes when the user only needs to go through the XML contents once.

Question 2

Air Bubble is a leading airline company and they have engaged your team to develop an online Airline Reservation System for them. Air Bubble has a main cabin seat in a flight. Passengers are allowed to choose multiple amenities such as Wifi, Live TV and head phones with the seat. Each amenity has its own cost associated with it. For example, if a passenger chooses Wifi and premium food, he/she will be charged for seat + wifi + premium food. The reservation system will show the price of the seat and the individual price of each amenity chosen by the passengers.

Identify an appropriate design pattern for the given scenario. For the pattern of your choice, provide:

- A brief description of the design pattern
- A class diagram to show how the design pattern may be implemented for the given scenario
- Justification for using this design pattern

(9 marks)

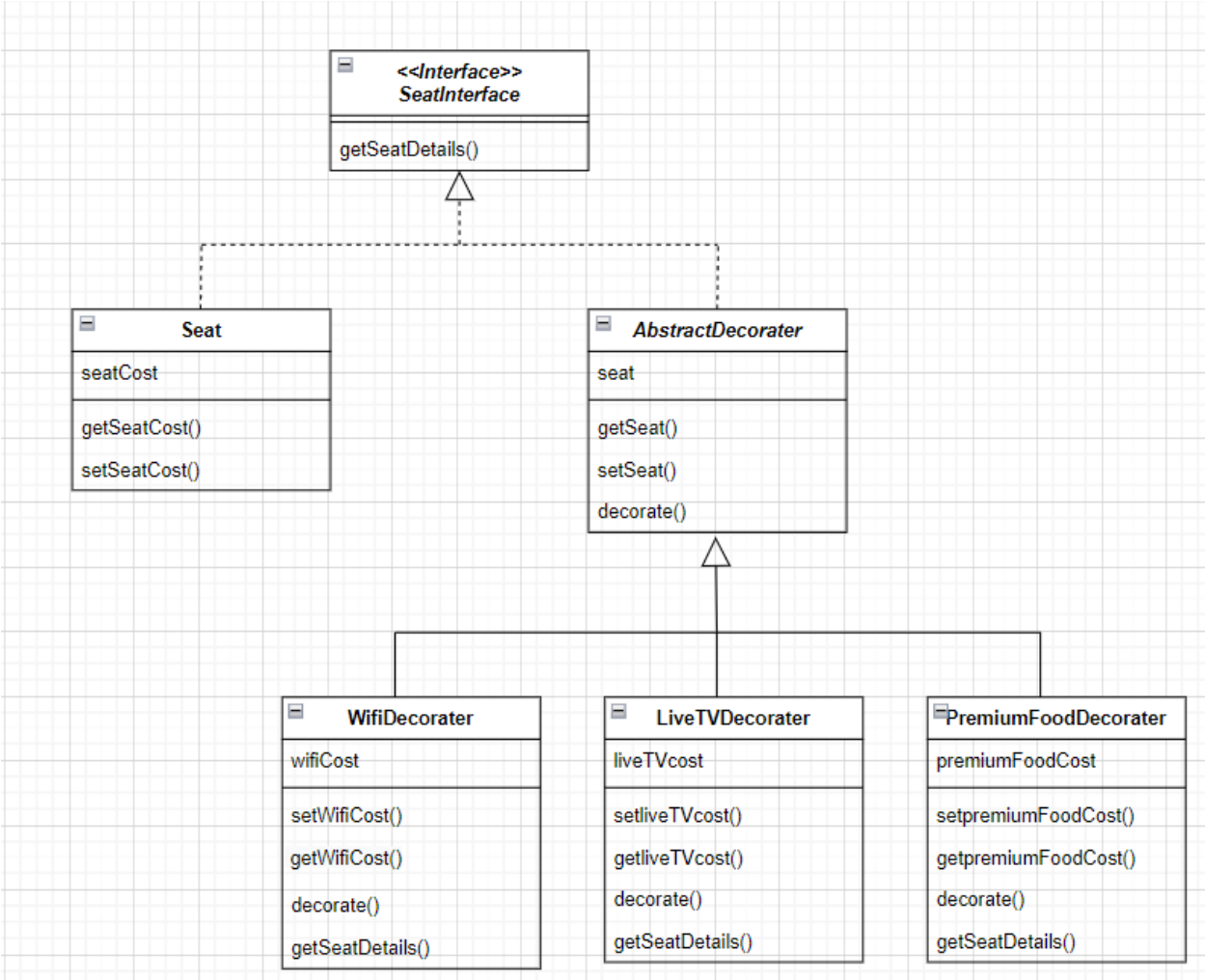
Appropriate design pattern

Decorator Pattern

Brief description of Decorator Pattern

Decorator is a structural pattern which provides the possibility of extension in its structure. It allows a user to add responsibility to an existing object transparently and dynamically without affecting its existing structure. To illustrate, the user can create a decorator class that wraps the original class and adds additional responsibility keeping the class method signature intact.

Class Diagram



Justification for using this design pattern

The reason why I choose decorator as the design pattern is because it allows us to add responsibility to individual objects dynamically and transparently. In this case, it allows the passenger to select one or multiple amenities to be added to the seat dynamically. They can either choose Wifi, Live TV, Premium Food or all to be added with their seat, the price will be updated accordingly. Moreover, the decorator pattern also allows us to withdraw the responsibility. For example, the passenger can unchoose the amenity chosen before and the price will be reduced from the initial seat price.

Suggest 4 secure coding practices for input validation. Your answer should include the rationale (基本原理) for each suggested practice.

There should be a centralized input validation for the application

- To ensure all the user input data is validated in a consistent way and provide a single point of maintenance which is easier to maintain. It's easier to trace data from entry point to exit point to know how the input is validated so that we can debug easily for correctness later.

Whitelist input validation

- Whitelist validation is the practice of only accepting input that is known to be good. This can involve validating compliance with the expected type, length, data range, or other format standards before accepting the input for further processing.
- This is to ensure that all the user input data is in the correct format before submit to the server and inserted into the database

All validation failures should result in input rejection

- When validation failure occurs, it should display a generic error message to indicate to the user that the input data is invalid and thus cannot proceed to the next step.
- It is to ensure that the genuine user does not submit invalid data to the server.

Encode data to a common set before validating (canonicalize)

- Canonicalization is the process of transforming a potentially flexible data structure into one that has guaranteed characteristics. It can reduce the software vulnerability if the input field has been canonicalized properly.

16 DEC 2020

Question 1

Case Study: Yum-Yum Restaurant

Yum-Yum Restaurant, a restaurant based in Klang Valley, has decided to provide online food ordering service for individual customers as well as corporate customers. At present, they already have a website that utilizes existing web standards such as HTML5 and CSS.

An appraisal of what led to the development of XML

HTML (hypertext markup language), the basic format for Web pages does not allow the definition of new text elements; that is, it is not extensible. XML adopts conventions that make it easy to parse, such as that document entities are marked by both a beginning and an ending tag, such as <BEGIN>…</BEGIN>. XML provides more kinds of hypertext links than HTML, such as bidirectional links and links relative to a document subsection.

TWO (2) reasons to support the use of XML in this use case

XML separate data from HTML

The data transported and stored in XML can be changed at any time without affecting the data presentation which is HTML markup language. To illustrate, HTML obtains data from XML and displays it on the web page, once the XML is modified or updated, it does reflect at the webpage without any changes need to be made in HTML code.

XML simplify data sharing

In the real world, computer systems and database applications normally are in incompatible format. XML stores data in plain text format, thus it makes it a system independent way of storing data and makes it easier to share the data among multiple applications.

TWO (2) practical examples to illustrate the usefulness of XML in this organization.

All the food and beverage information can be extended and modified easily by putting them as XML format. For example, we can add on promotion deals about the special occasion food and beverage easily for incoming Hari Raya . Comparing to database, it will be more tedious as the user will need to create a new table specially for promotion or insert more column to insert the promotion details

The XML document that consists of menu information can be easily merged with style sheets to create desired output and presented in an attractive way to the customer. For example, all the menu information can be displayed in tabular table format that looks organized to the customer for ordering purposes.

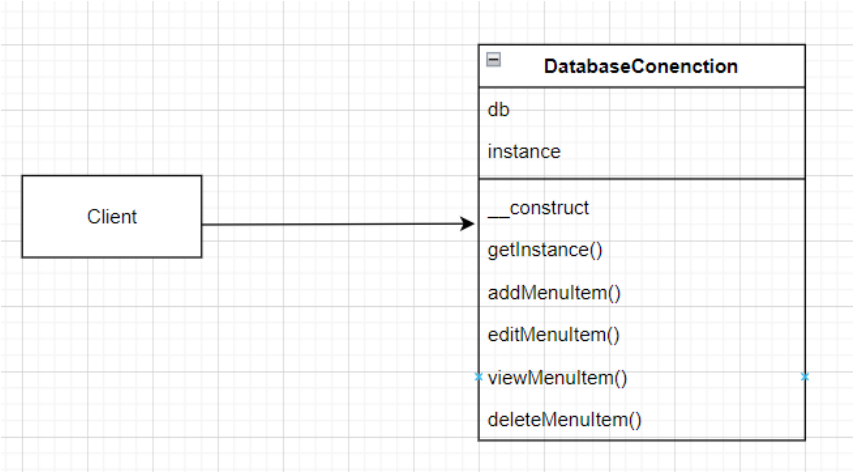
Question 2 (a)
Select TWO (2) design patterns that would be appropriate for Yum-Yum's proposed online ordering system. Compare and contrast these two design patterns in terms of the following aspects:

- The purpose of using the design pattern, and
- Class diagrams depicting how the pattern will be applied in the proposed system.

Design pattern
Singleton pattern, a creational pattern to ensure an object can have only one instance and provide global point access to it.

Purpose
When the user wants to have exactly one interface for database connection in Yum-Yum's online ordering system. If multiple database connections exist, it will be expensive and might not be supported well by the system. Hence, the user wants all parts of the system to share and access one database connection.

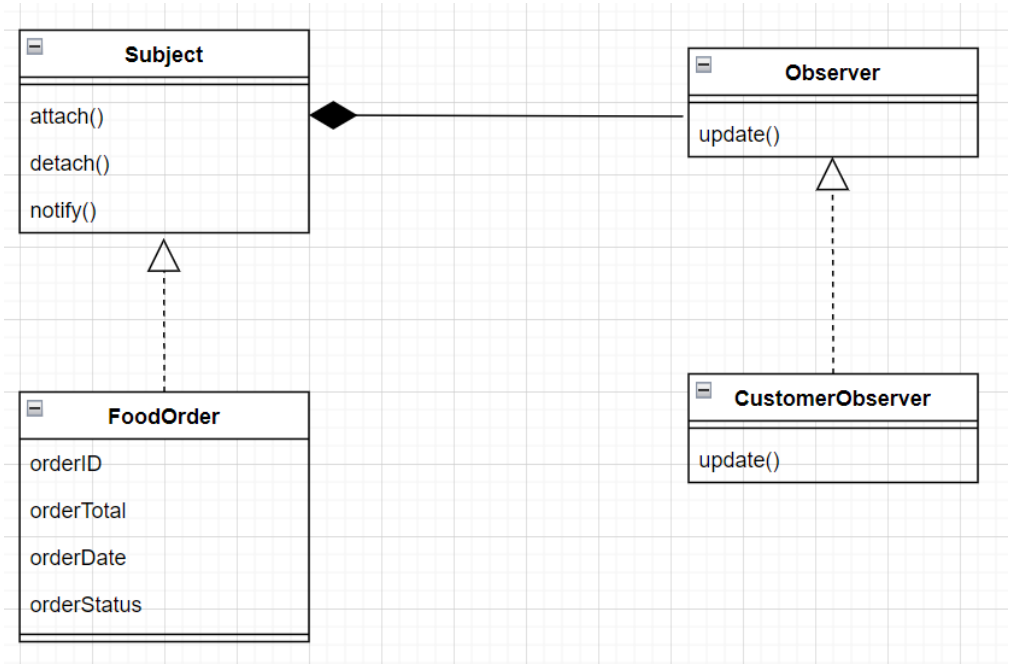
Class Diagram



Design pattern
Observer pattern, a behavioral pattern refers to one-to-many dependency in which when its subject changes state, the dependent object will be notified and updated automatically.

Purpose
Observer pattern is used in Yum-Yum's online ordering system for order tracking in real-time. Whenever the food ordering status state changes such as from "Preparing your food" to "Your delivery is on the way", "Food delivered", "Food canceled" etc, the responding customer observer will be updated and notified automatically according to its subject (Order object) state.

Class Diagram



Another observer example: Quantity on hand, low stock, alert

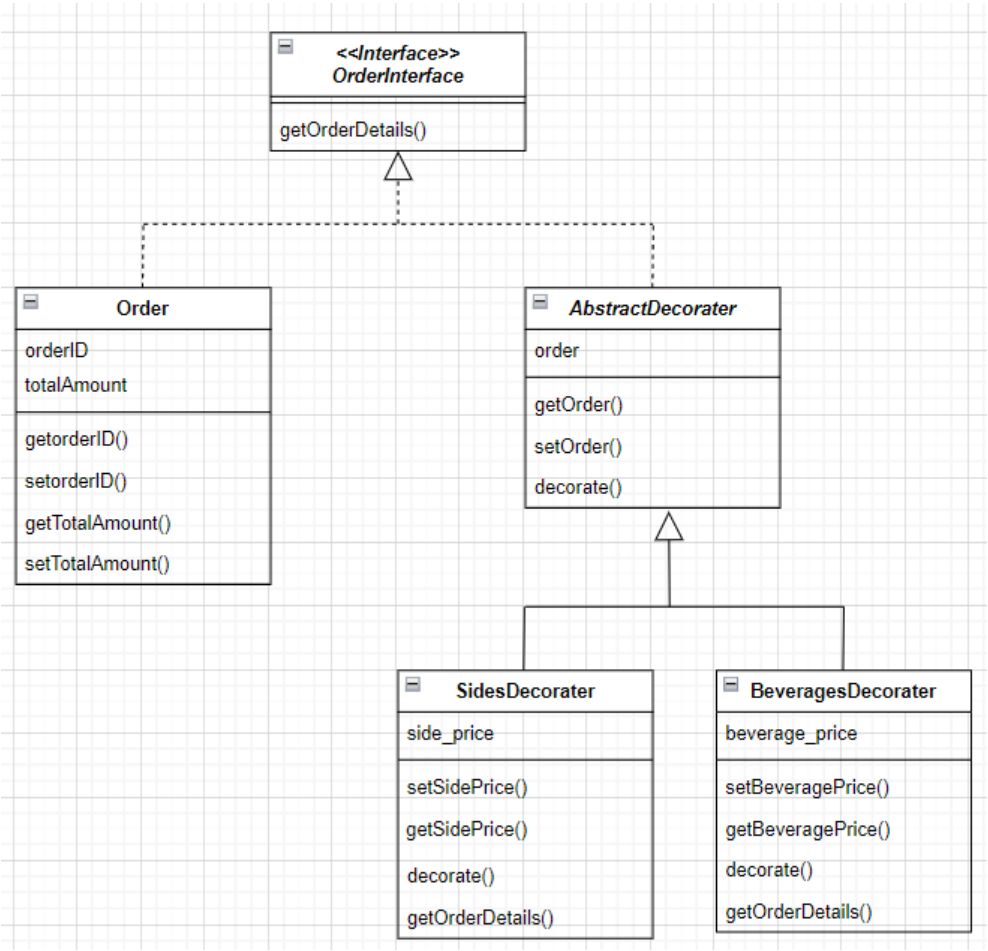
Design pattern

Decorator pattern, a structural pattern to add additional responsibilities to the existing object transparently and dynamically.

Purpose

By applying a decorator pattern, it allows customers to add on food item add-ons to the food order dynamically. They can choose the available add-ons to be added with their food order, the total amount will be updated accordingly. Moreover, the decorator pattern is suitable when the responsibility added can be withdrawn. For example, the customer can as well unchoose the add-ons chosen before and the price will be reduced from the food order total amount.

Class Diagram



Question 2 (b)

Differentiate between the following pairs of terms. You should provide detailed explanations and also include suitable examples based on Yum-Yum Restaurant's context in your answers.

(i) Scripting languages and application languages

Scripting language	Application Language
A Scripting Language is a programming language that is mainly used to **automate certain tasks within a software**	An application language consists of instructions for the computer and is used for **creating complete software application**
There is no need for compiling the script, instead it is directly **interpreted at runtime**	The program is **compiled by the compiler** at the execution time
There is **no executable file** generated during execution of the script.	An **executable file is generated** during code execution
Tend to be **weakly type** in which variables are	Tend to be **strongly type** in which variables are

declared without specifying the data types	bound to specific data types
Scripting languages are used to **create dynamic web applications**	Application languages are used to **write computer programs**
They are usually **small pieces of code**	The code is usually large and has a **large number of lines**
Coding in scripting language **encourage rapid prototyping and development** as the script can be used as software prototype thus saving time on testing project	Coding in application language is **time taking** as it involves designing a complete software
Examples: **PHP, JavaScript, Python**	Examples: **C, C++, JAVA**

(ii) Shell scripting and extension languages

Shell Scripting

It glue together as a sequence of OS commands to be carried out. It is a computer program designed to be runned by Unix Shell, a command line interpreter. Shell scripting is used for system level operations such as automating tedious repetitive tasks, automating the installation process for new software or new software updates across the organization. In Yum-Yum's online ordering system, the developer can use shell scripts to develop system-level scripts that will run during system boot-up such as develop scripts to automate the new dependency package installation of the online ordering system. By using shell scripting, the development team can increase the speed of the development process of an application. The examples of shell scripting languages are bash and sh.

Extension language

It is used to extend and control the application. This means that the developer can "easily'' connect one language with another language in an application. For example, the Yum-Yum's online ordering system initially is developed using only HTML and CSS to display the output to the customer, the developer can use the Javascript as an extension language to perform client-side validation at the check-out form or connect API with Javascript to fetch or give the information to the server.

18 SEPT 2018

Question 1 (b)

Provide a detailed comparison in terms of the way XML documents are parsed using Simple Api for XML (SAX) and Document Object Model (DOM)

SAX Parser

- Event Driven Parsing
- Serial Access Traversal - XML is read sequentially
- Read-only API
- When a parsing event happens, the parses invokes the corresponding method of relevant handler
- The handlers are the programmer's implementation of standards API such as interfaces and classes

DOM Parser

- DOM parser create an internal data structure which is a tree object based on the XML document
- The entire XML document will be stored in memory once the tree has been constructed
- User can access the XML data by traversing the tree in any direction
- Read-Write API - The API allows for constructing, accessing and manipulating the content and structure of the XML document

Question 2

Natale Trees is a company that supplies Christmas trees and they have engaged your team to develop an online Christmas Tree Ordering System for them. Besides the basic Christmas tree, customers can choose to add on decorative items such as Christmas bells, baubles, star ornaments, tinsels, rope lights, and so on. After ordering the tree and decorative items, the system should show the price of the tree including the individual price of each decorative item chosen.

Identify an appropriate design pattern for the given scenario. For the pattern of your choice, provide:

- A brief description of the design pattern
- A class diagram to show how the design pattern may be implemented for the given scenario
- Justification for using this design pattern

(9 marks)

Design pattern

Decorator pattern, a structural pattern which provides possibility of future extension in its structure. It allows for adding additional responsibilities to the existing object dynamically and transparently without affecting the structure. It create a decorator class that wrap the original class and add additional responsibilities keeping class method signature intact

Class Diagram

Refer to 12 February 2019 [They are similar, just change the name of those classes]

Justification of using Decorator Pattern
Refer to 12 February 2019 [They are similar]

Question 2 (c)
Suggest 4 secure coding practices for password management. Your answer should include the rationale for each suggested practice

Enforce the password complexity and length requirements to be specified in policy and regulation

- This makes the password difficult to be cracked by malicious actors as the more password requirements we enforce, the higher possible password combinations of letters, digits and symbols thus make it harder for hackers to guess.

The password entry should be obscured from user screen

- To protect password and prevent password leaked issue, which is, the password that remain in a clear text format will be easily stolen by malicious actor to pretend as a legitimate user and access your account

Account disabling after a limited number of login attempts

- Prevent brute-force attacks as the hacker can automate this kind of attack to try thousands or even more of password combination to get access to the user account and perform malicious actions

If using email-based reset, only send email to a pre-registered email with temporary link that has a short expiration time

- This prevents the password from being reset by a malicious user. If the email getting hacked, a temporary link with short expiration time ensure the hacker cannot use it to reset the password as it already been expired and not working anymore

1 OCT 2020

Analyze RealKozy Furnitures' proposed information system and determine TWO (2) design patterns that would be appropriate. For each design pattern, draw a class diagram to illustrate the implementation of the design pattern in the proposed system.

Note: You may make any reasonable assumptions.

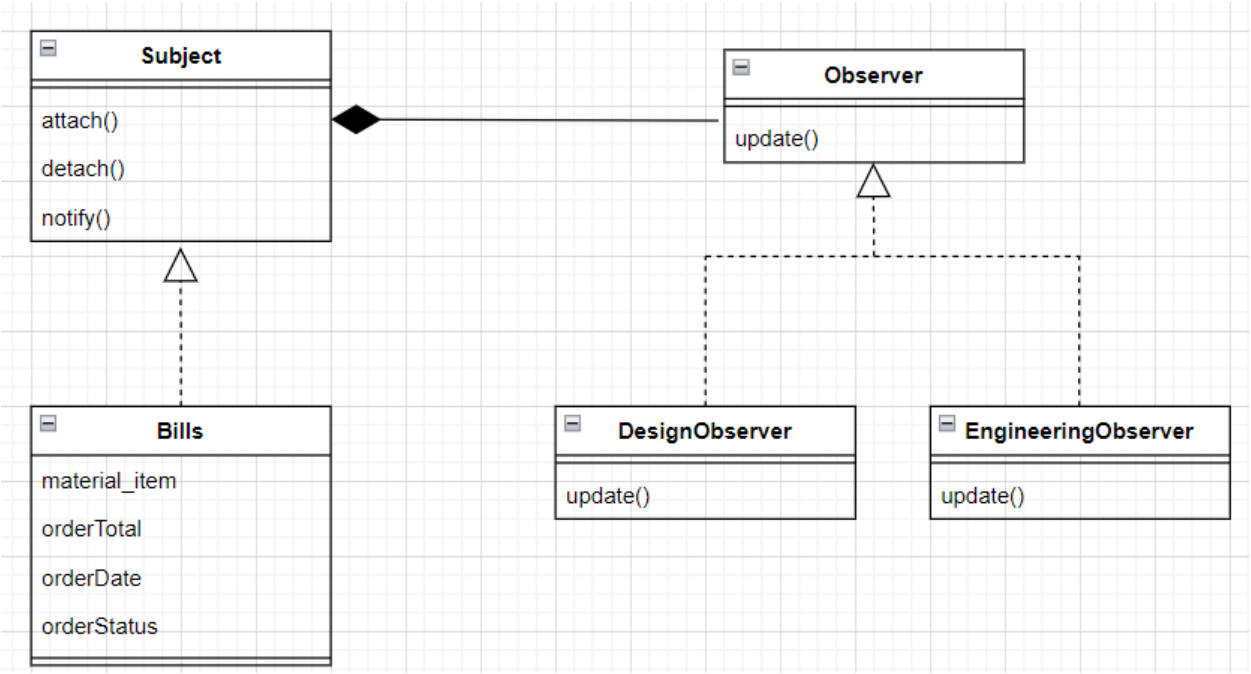
Answer

Design Pattern

Observer Pattern. It is a behavioral pattern and has one-to-many dependency in which when a subject changes its state, the dependent objects will be updated and notified accordingly.

Observer pattern is suitable when changes in one object will result in another changes in other objects and you don't know how many objects need to be changed. In this case, each bill of materials (we don't know how many materials need to be ordered) will need to be updated frequently, so the design and engineering changes can be reflected depending on the bills.

Class Diagram

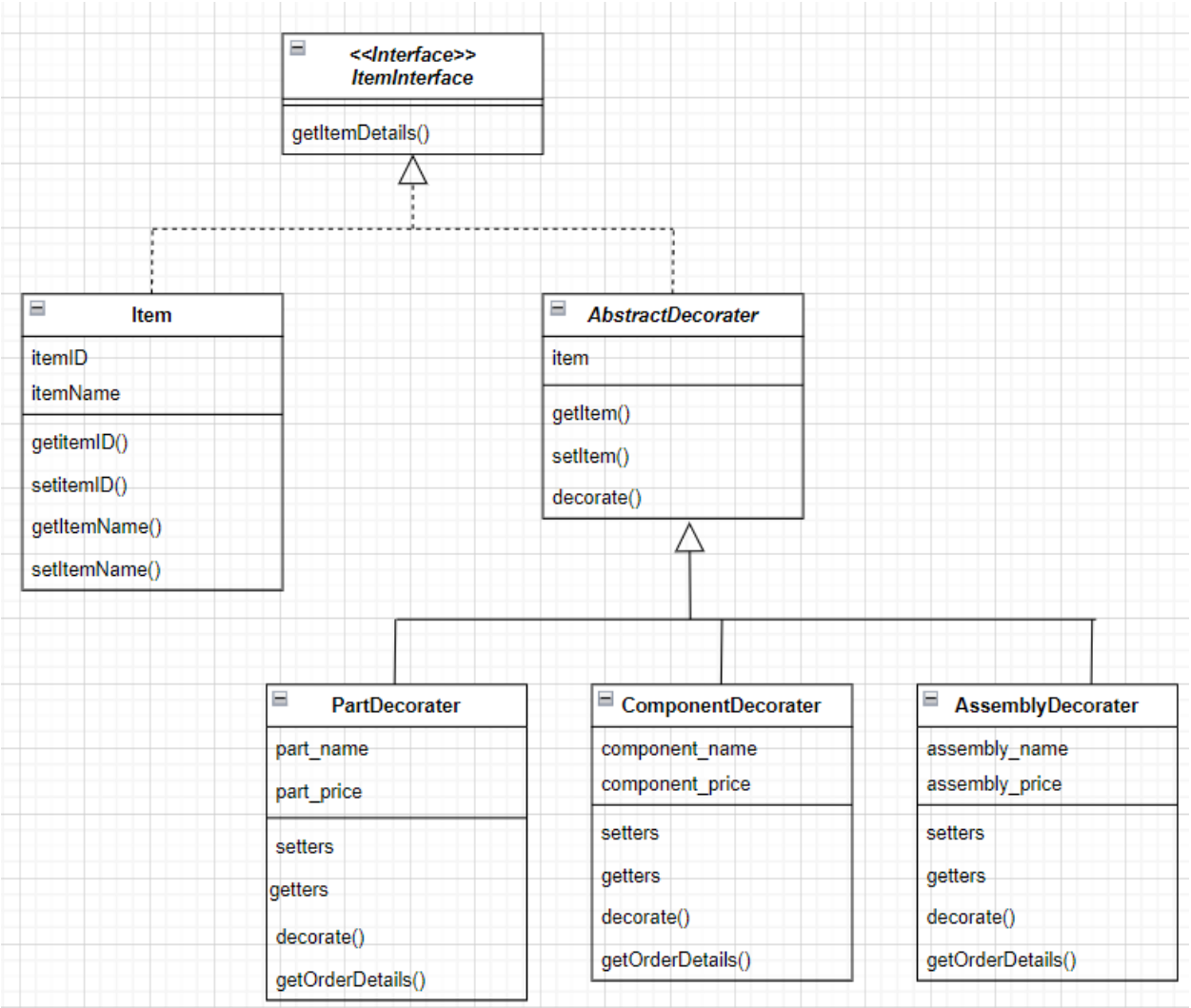


Design Pattern

Decorator pattern. It is a structural pattern that allows for adding additional responsibilities to an existing object transparently and dynamically without affecting the structure. In this case, the item is made up from different parts, components, and assemblies which is suitable for

applying a decorator pattern. It allows the manufacturer to add the necessary components needed to produce the items.

Class Diagram



Question 1 (b)
Appraise THREE (3) general benefits that design patterns bring to RealKozy

Facilitate reuse
Reuse design patterns can prevent subtle issues that can cause major problems as the reusable code that uses the design pattern has been proven as safe, secure, and reliable. It also improve code readability for developer who familiar with the design pattern

Define a shared vocabulary for discussing design
It makes the communication between developers become more efficient, which is, the developers can immediately think of the high level design in their mind when they hear the

name of the pattern during discussion. This saves the time as they do not need to research more about the design pattern and straight away can jump into the point for system development.

Demonstrate principle and concepts of a good design
The classes created are reusable and extensible. Hence, it can speed up the development time and ensure compliance standards achieved as the class has been reused many times and proven to be high code efficiency and quality.

Question 1 (c)
General comparison

With Design pattern	Without Design pattern
No need to reinventing the solution strategy	Need to reinventing the solution strategy from scratch
Save development time	Slow down the development time
Low development cost	High development cost

Question 1 (d)
"The object-oriented programming paradigm is more superior than the procedure-oriented paradigm."

Do you agree with the above statement? Provide reasons to support your opinion. (6 marks)

- Yes, I agree with the above statement.
- The OOP supports four basic concepts which are abstraction, encapsulation, inheritance and polymorphism that the PP does not support.
- The abstraction of OOP paradigm provides data hiding thus it is more secure compared to procedure-oriented programming which does not have a proper way to hide the data.
- Moreover, the OOP can make the complex logic hide behind and the system easy to use for a user as they do not need to know how the class is implemented.
- The inheritance of OOP lets the developer reuse the code thus saving the development time as well as improving quality as the code has been debugged and reuse proven.
- Compared to procedure-oriented programming, it is hard to maintain as if a sub-procedure has been modified, it becomes difficult to find and maintain it. Eventually it will increase the development time.
- Hence, the OOP is more efficient and superior than the PP

Question 2 (a)
Show how the following secure coding practices may be applied to ensure there is an appropriate level of software security in RealKozy Furnitures' information system by giving specific examples and providing detailed elaboration for each answer:

- There should be centralized input validation for the system
- Implement the principle of least privilege
- Use whitelists

There should be centralized input validation for the system

- The data can be validated in a consistent way and it provides a single point of maintenance.
- It's easier to trace data from entry point to exit point to know how the input is validated so that we can debug the code easily for correctness later.

Implement the principle of least privilege

- Manager access right vs normal employee access right

Whitelist input validation

- Whitelist validation is the practice of only accepting input that is known to be good. This can involve validating compliance with the expected type, length, data range, or other format standards before accepting the input for further processing.
- This is to ensure that all the user input data is in the correct format before submit to the server and inserted into the database
- Check out the stage when performing furniture ordering by the customer. [describe it]

Question 2 (c)
Propose TWO (2) appropriate web services that may be published by RealKozy. Provide a description of the web services and examples of how they may be consumed by other applications. [10m]

Notes before answering this question
Two web services that may be published - That means what services the company can provide Similar to what you did in your assignment. What web services that you had published? Like a list of hotel room information, etc. It is referring to two examples of services.

Description of the web service
To provide a list of furniture products' details based on the search keyword(s).

Input parameter

- SearchInput that represents the furniture name from the search container.

Output

- This method will return the furniture products' details based on the keyword which is the furniture name.

Example on how may be consumed

This may be used by applications in an interior design company that would like to provide quotations based on the furniture items that are included in the proposed design. This could also be used by furniture retailers that are considering the collection of furniture to be included in their next purchase.

Description of the web service

To provide furniture ordering functionality in another application
Input parameter - Customer Information and order details
Output - This method will return the successful message about ordering and order receipt

Example on how may be consumed

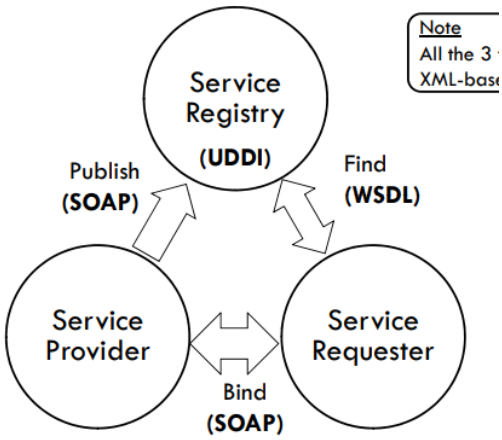
RealKozy will be integrated with an application called EasyBuy that provides a buying and selling platform for customers and furniture shops. As part of the partnership, the EasyBuy customer will be able to order furniture items from the EasyBuy app. Afterward, RealKozy will start to produce the furniture and shipped to that customer.

Note only if the question mentions **type of communication protocol** or **web services category**, then we need to describe SOAP or REST

SOAP (Simple Object Access Protocol)

There are 3 essential technologies that included in SOAP which are SOAP (Simple Object Access Protocol), WSDL (web service description language) and UDDL (universal discovery description integration)

SOAP - A protocol for exchanging the XML document, typically using HTTP
WSDL - Describe the description language interface for web services
UDDI - A system that enable client and server to find each other



How to consume UDDI

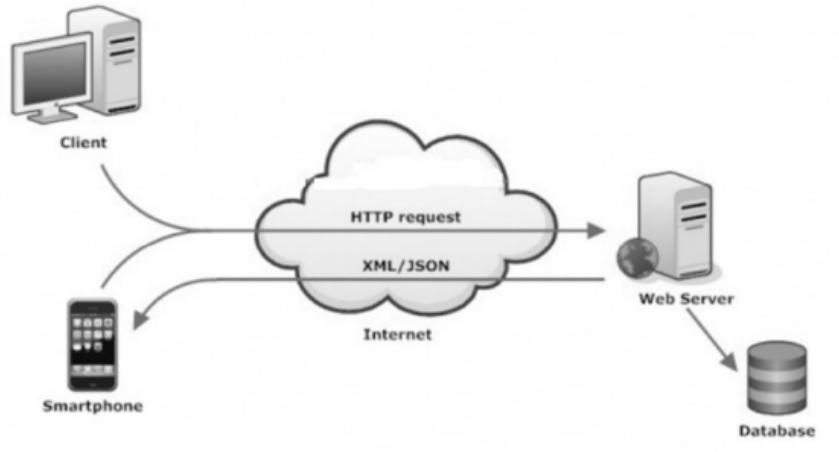
EXAMPLE: USING UDDI

- If the industry published an UDDI ***standard*** for flight rate checking and reservation, airlines could register their services into an UDDI directory
- Travel agencies could then search the UDDI directory to find the airline's reservation interface
- When the interface is found, the travel agency can communicate with the service immediately because it uses a well-defined reservation interface (by WSDL)

REST (Representational State Transfer)

- It is a set of principle rather than protocol
- In RESTFUL services, 4 HTTP verbs are used to perform basic CRUD functionalities which are GET, PUT, POST and delete.
- RESTFUL service deal with transferring representation of resource
- The representation can be XML, JSON, etc

REST WEB SERVICE



How to consume API

- We can consume the REST API resource (its application functionality that is represented by a unique URL) by sending a HTTP request (PUT, GET, POST, DELETE) from the client to the server to get/give the information from/to the server.

26 April 2018

What are web services? (1m)

- Web service is a technology that enable one program to communicate with another program through HTTP over the network
- Web service enable a program on one system to invoke method at another system
- Web service is a self-describing, self-contained software module available via the network

The Combo SOAP, WSDL and UDDI defines a general model for web service architecture. Discuss the purpose of each technology. (6m)

SOAP - A protocol for exchanging the XML document, typically through HTTP
WSDL - An interface description language about web services
USSI - A system that enable client and server to find each other

There are two types of XML parsers which are SAX and DOM. Discuss how these parsers are different. (6m)

SAX

- Event-based parsing
- Read-only API
- Read XML document sequentially - Serial access protocol
- When a parsing event happens, the parser will invoke the corresponding method of relevant handler
- The handler is the programmer implementation of standard API such as interfaces and classes

DOM

- Create a internal data structure (i.e tree object) based on the XML document
- User can access the XML data by traversing the tree in any direction - Random access protocol
- The API allows for constructing, accessing and manipulating the content and structure of the XML document - Read-Write API