Reflection Questions on assignment 1/2/3/4

1. What is the relationship between containment and encapsulation?

In this project, OptionSet object is created in Automobile class, Option object is created in OptionSet class, these are considered as **containment**.

Also, the containment helps OptionSet and Option object be encapsulated by providing getters, setters and other methods. **Encapsulation** is a safer way, through which the instance variables are not exposed to outside users, and we can do some change for our project later, without breaking anybody else's code.

2. What are some ways to analyze data to design objects?

One way is reading lines from a text file and setting the variables for objects. In this way we need **BufferedReader**. Wherever we need to use BufferedReader, we must declare it in a try statement and close it at the end.

The other way is reading properties file and setting the variables for objects. In this way we need **FileInputStream** and **Properties object**.

3. What strategies can be used to design core classes, for future requirements, so that they are reusable, extensible and easily modifiable?

For the class Automobile, We need to consider the encapsulation; make the instance variable private, make the method public.

For the classes like OptionSet and Option, we make the instance variable private, and make the method protected. Protected functions prevent themselves being invoked by classes from other packages. In order to access the protected functions, the outer class(Automobile) should implement the public functions and be in charge of communicating with other classes.

4.What are good conventions for making a java class readable?

If we want to use the components inside other classes, the **import** statements are very useful.

There are two kinds of import statements:
a. Specific import. e.g. import model.Automobile (in in FileI/O class)
b. Wildcard import. e.g. import java.io.* (in FileI/O class)

5.What are the advantages and disadvantages of reading data from sources such as text files or databases in a single pass and not use intermediary buffering?

The advantages of reading data from text files or databases in a single pass and not using intermediary buffering is that it can reduces I/O overhead and save memory for saving temporary data.

The disadvantages is that it is more difficult to parse the data because we have no ideas about how much data is in the resource hence we are unable to initial the size of data structure before hand if the size is not specified in the resource.

So the point of using intermediary buffering (BufferedReader) is that it can make data reading more efficient.

6. What is the advantage of using serialization? What issues can occur, when using serialization with inner classes?

Serialization is an easy and safe way to save and restore our objects. Serialized object can be transferred via network because Java serialized object remains in form of bytes which is portable.

When using serialization with inner classes, we must make sure that the inner class implements the Serializable as well as the outer class does. Otherwise, we can’t do the serialization successfully.

7. Where can following object relationships be used: encapsulation, association, containment, inheritance and polymorphism?

Association is a general binary relationship that describes an activity between two classes. For example, each model (Ford Wagon ZTW) definition has a set of properties (Color, Transmission etc.), that is an association between the Automobile and OptionSet. We initiate OptionSet array list in Automobile, then “the list contains OptionSet objects” by implementing setter methods.

We can use containment/encapsulation under this condition: classes in one package, using import statements.(see the question1)

We use inheritance/polymorphism under these conditions: subclass extends superclass, and classes implements interface.

8. How can you design objects, which are self-contained and independent?

First, when designing objects in java, we will take the concept “Object Oriented Programming” in mind.

There is a structure in Java called class, which let us construct our software in a particular way. Using classes, we can construct objects. Each object made from that class can have its own values for the instance variables of the class. And objects can call methods on one another. So an object is nothing but a **self-contained** component which consists of properties and methods. We can ask the object to do something, through invoking one of its methods.

The other thing we should consider in our designing is: Java is **platform independent**, JVM is platform dependent.

Platform independent means that the java source code can run on all OS(operating systems), and the process is called “compile”.However JVM depends on the OS: if you are running Mac OS you will have a different JVM than if you are running Windows or some other OS.

9. What role does an interface play in building an API?

In lab2, we build an API containing interface classes(CreateAuto and UpdateAuto), abstract class(ProxyAutomobile), and subclass(BuildAuto). Interfaces are exposed to users, and give users controlled access to the complex implementations. They enforce specific class(ProxyAutomobile) implementing all the methods.

Interface class is like a bridge connecting internal and external application.

10. What is the best way to create a framework, for exposing a complex produce, in a simple way and at the same time making your implementation extensible?

In lab2, we build an empty class(BuildAuto) which implements all **interfaces** and extends ProxyAutomobile(so it gets all the implementation of interfaces methods). Through this way we can hide our code inside the ProxyAutomobile and stop users getting into our program.Interfaces are a way of decoupling separate software components.

11. What is the advantage of exposing methods using different interfaces?

When designing interface class, we should not put all methods in one interface, instead we split this interface into smaller and more specific ones. For example, we create CreateAuto which has BuildAuto and printAuto methods in it , as well as UpdateAuto containing updateOptionSetName and updateOptionPrice.

This is called **interface-segregation principles**, which states that no clients should be forced to depend on methods they are not interested in or do not need to use.

12. Is there any advantage of creating an abstract class, which contains interface method implementation?

ProxyAutomobile is the **abstract** class, which contains interfaces methods implementation. ProxyAutomobile uses code from internal component therefore it is integrating internal and external components. Everything we do not want to exposed to the client should be in this class, and it is abstract to limit the client from instantiating it.

13. How can you create a software architecture, which addresses the needs of exception handling and recovery?

In this project, AutoException is a custom exception class. The custom exception all been throne with an AutoException object with a specific error number. When the AutoException is caught, the fix method for that object is called. Based on the error number the fix method calls a specific method that can fix the exception or just report the detail of the exception. At the same time a log method is called that logs the information of the caught exception for future analysis.

14. What is the advantage of exposing fix methods for exception management?

AutoException extending Exception is used to fix exceptions. We expose this class with an interface FixAuto. This is the consideration of extensible design. (see question10)

15. Why did we have to make the Automobile object static in ProxyAutomotive class?

We use a static list, **LinkedHashMap**<String, Automobile>, to be able to share it between different instantiation of BuildAuto by different interfaces.This is a way to achieve one single source of truth.This linked hash map is a **singleton**.

A singleton should be used when managing access to a resource which is shared by the entire application, and it would be destructive to potentially have multiple instances of the same class.

16. What is the advantage of adding choice variable in OptionSet class? What measures had to be implemented to expose the choice property in Auto class?

In OptionSet class, choice variable is private and set/get choice methods are protected. In order to access the protected functions, the outer class(Automobile) should implement the public functions(set/get/find choice methods)and be in charge of communicating with other classes.(see question3)

17. When implementing LinkedHashMap for Auto object in proxyAuto class, what was your consideration for managing CRUD operations on this structure? Did you end up doing the implementation of CRUD operation in proxyAuto or did you consider adding another class in Model for encapsulating Auto for the collection and then introducing an instance of this new class in proxyAuto.(Think about this and if this part of your design is not self-contained, then fix it)

CRUD is the abbreviation of: Create, Read, Update, Delete.(consideration about ProxyAutomobile and LinkedHashMap, see question12/15)

18. What is the best way to setup multithreading in an Enterprise Class application?

Enterprise application indicates that the application platform would be too large and too complex for individual or small business use. Using synchronization is the best way to setup multithreading in this kind of application.(*)

19. What strategy is used for synchronizing, so you end up with a scalable application?

When we applying synchronizing for multithreading, we do not synchronize all the variables or methods of threads we created in EditThread class, instead we just synchronize the methods updateOptionPrice( ) and updateOptionSetName( ) which are invoked by threads.

20. What implementation strategy can be used for creating a race condition for testing Multithreading?

Race condition means two or more threads have access to a single object's data. Race conditions lead to data corruption. So the simple way is making two threads and editing the same property of one object.

21. How does Synchronization work in JVM? What are the performance consequences of Synchronizing?

Every java object has a lock. A lock has only one key. Most of the time, the lock is unlocked and nobody cares. When supplying synchronized methods to an object, a thread can enter one of the synchronized method only if the thread can get the key to the object's lock. The role of JVM is helping the thread finding the key.

When code hits a synchronized method, there's going to be a performance hit, which is not noticeable. For large amounts of data, the synchronization process needs to be carefully planned and executed to avoid any negative impact on performance.