Name : Nianthrini Vivekanandan
Class and Section : CIS 35B De Anza -Java programming
Assignment Number : Lab 6
Due Date : 06/20/2016
Date Submitted :06/20/2016

LESSONS LEARNT DOCUMENT

1. To design core classes, for future requirements, so that they are reusable, extensible and easily modifiable. Think about the future implementation of the project/classes/methods and design your project accordingly. This is called designing ahead and increases productivity. In addition, CRUD (Create, Read, Update and Delete) operations have to be considered when creating getters and setters and would be helpful for designing ahead. If for the automobile class, only the given properties, values and prices are designed then changing the name, value and prices as well as adding new properties would be extensive. Instead if classes are designed to isolate the property which can be resused for any property would make changes and additions much easier. CRUD operations also can be added which would help in changing or adding or deleting properties for future use, reusable, extensible and easily modifiable. Also, it is beneficial to organize the classes in functionally relevant packages so that related classes can use protected variables and methods within the packages.

2. Top-down approach is the process in which the design is built based on what is only needed. Bottom-up approach is the process in which the design is built keeping the future in mind. Start with the lowest entity and build methods from there going up to the largest entity.

3. Do not start creating classes for each option and option set. This would only work for the particular car model. Try to recognize the pattern and design the classes in such a way that it would work universally for any model being designed. Here, each automobile has a price, name and properties. Each property has a value and price. Each property has one or more values and each value has a price. So, if we design a class to represent property and another class to represent the property value and price, it can be reused for every

property of automobile. It is important to design the classes before starting the coding process.

Classes can also be designed based on it usage. Here, we need a class to read the input file and parse the values for the properties, values and prices. All the functionalities to read the file are put together in a class for FileIO.

After analyzing the needed classes, the methods to be added for ach class is designed. Constructors of every possible meaningful combinations have to added, methods to get and set the private variables are also to be added, then methods to get different properties, their values and prices are to be added taking into consideration the CRUD operations.

4. Options and option set class have private variables and protected methods to encapsulate the methods from other packages. These 2 classes can only be assessed via Automobile class and prevents unauthorized changes in the 2 classes. This is encapsulation.

5. Options were made an inner class in the option set class. This would prevent any other class except option set class to access the methods and variables of the options. Inner class is hidden from all other classes including those in the same package. Thus, automobile class can access options only through methods of option set class. This is also encapsulation but with containment which is a class-level containment.

6. There are three types of encapsulation – Object, class and containment. Class level encapsulation is done by declaring the instance variables private/protected. Object level encapsulation is done by declaring the instance object of the class as private/protected.
Containment level encapsulation is done by declaring a class as inner class.

7. StringBuffer has to be used to concatenate strings. This is because strings are immutable and waste lot of memory compared to StringBuffers.

8. Do not use static variables unless you are dealing with singleton. This is because static variables are global variables and cannot work if you want to work with different instances and can cause problems during

multithreading if you need 2 different objects of the same class. So, static variables are good to use with universal constants like pi and singleton but generally avoid static variables as it is not object oriented design which deals with object instances and static variables can be used without instance objects.

9. When an import for a package is made, it does not automatically import all its sub-directories. Sub-directories have to be imported manually.

10. When reading a file, the whole file has to be read without storing the values in an array or other collections, the storing of values has to be done directly into the variables of the class object. This is because using any buffer may require additional memory and time. If Strings are used for storing variables, each time the string is updated, memory space is wasted as Strings are immutable.

11. Serialization is the method to write data to a file. The object which lives in the computer's memory is written to the disc using serialization. Inner classes do not inherit serialization. Sub-classes inherit serialization from the super class.

12. Interfaces are being used to control access to various functionalities which would otherwise be available to the end-users. This is also encapsulation using interfaces.

13. BuildAuto class is written as empty class without any functionality for security. If the classes were back-engineered, having BuildAuto class as empty will prevent access to functionalities as the methods are defined in a proxy class. Delegating functionality to a proxyAutomobile class from BuildAuto is called value proposition design pattern in OOP.

14. The delegating proxy class is made abstract to provide artificial security. If the classes were back-engineered, having an abstract class will prevent instantiation of the class preventing hacking or access to functionalities.

15. All the healing can be centralized in one place using custom exceptions.Exceptions can be fixed in-line where errors occur or it can

be propagated to the calling class. Exceptions can be thrown in 3 ways - using constructors of AutoException class, using if, throw and catch and using while loop and catch. Machine learning is implemented using custom exceptions where exceptions are fixed by learning about the different exceptions that can occur.

16. Making FixErrors an API would allow users to call fix methods based on the error number. This is also helpful for software testers who can fix the errors by calling the methods corresponding to the error numbers.

17. Making the automobile/fleet object a singleton or static variable is important to keep the automobile object/ LinkedhashMap of automobiles intact across the whole program and helps to integrate the functionality across different classes.

18. An arraylist of options in the automobile class stores the option choice made by the user for each auto model. This help to get the price and name of the chosen options instantly in the auto class without using the optionset or options class to get to the values.

19. Option class is eventually changed to a regular class from inner class. This is because inner class leads to spaghetti code and there is a requirement for option to be imported in automobile class or it can become a regular class to be accessible in automobile class. Inner class also does not inherit serialization from the outer class which is another disadvantage.

20. Arrays are not recommended for use for variables which do not have a fixed size. Arraylists can allocate memory dynamically and does not have a fixed size as opposed to Arrays and can give a user flexibility when changing data in the variables. If an array has to be changed, a new array has to be created and the old array copied to the new location. But Arraylists can dynamically change it's size. To add elements to arrays we have to keep track of indices which is not needed in arraylists.

21. BuildAuto is made final to prevent extending of the class. This would prevent unwanted usage of methods defined in BuildAuto and adds security to the code.

22. The use of templates enable any type of vehicle to be modeled using this program. To develop templates, automobile was declared abstract and the linkedhashMap object of autos is the variable in the fleet class which acts as a template. Declaring automobile class as abstract would prevent it's instantiation as an object since it would not be self-contained once a fleet of automobiles are created.

23. ProxyAutomobile class acts as a bridging class between the interfaces and BuildAuto class. All the method implementations of interfaces of BuildAuto are done in ProxyAutomobile. This is advantageous in encapsulating the actual implementation from end users as well as preventing hacking of the application by back-tracing the class definitions.

24. Using Fleet class with a LinkedHashMap of a fleet of autos enables instantiation of more than 1 auto object. Thus, the application can handle more than 1 auto object and with further additions can also handle multiple users.

25. In Object Oriented Design, changing one thing should not disrupt the structure of the other components.When building an application, the functionality of methods which was used in the older versions should not be changed, but new functionalities and methods with different signatures are to be added. This is because if the old functionality is changed, a client running an older version of the product would not be able to run the application correctly. So, it is important to make sure that the first version of the application runs exactly as intended even though updates or functionalitites are added to enhance the application.

26. Implementing multithreading with synchronization in an application is important to make it scalable because without it, the data can get corrupted if multiple threads without synchronization try to change the value of a single object at the same time. It is important to keep the threads synchronized so only 1 thread changes the value at a time. Multi threading without synchronization would allow 2 thread to access the same value at the same time and can result in race condition where the change made by first thread would be overridden by the second thread.

27. To enable multi-threading with synchronization, it is important to lock the object on which the threads are acting. It is important to create a lock on the smallest entity possible that the thread would act upon. This is to allow threads acting on other objects to continue running without any interuption. If the whole fleet of linkedhashmap is locked, other threads cannot change any value at all and would stay idle until the lock is released.

28. It is also important to use, wait() and notify() during synchronization. A condition check can be made to see if another thread working on the same object is running, if it is running, wait for the thread to complete before another thread can be run. Once, the thread completes it's run, it can notify the other thread and falsify the condition, this releases the lock on the object and the other thread can modify the object.

29. EditOptions and ProxyAutomobile have a 2-way relationship, where EditOptions uses the methods of ProxyAutomobile and ProxyAutomobile uses EditOptions to implement multi-threading. The interface is used in the driver class, whose methods are defined in the ProxyAutomobile, which in turn instantiates an EditOptions object and starts the thread. The thread method uses the methods from the ProxyAutomobile class. Thus, it is a 2-way relation.

30. Two projects have to be created to handle the client and server side interaction. This is because only 1 main can be run from 1 project. Also, eclipse has multiple consoles for ouput.

31. File reading functionalities have to be written to read both user-friendly as well as engineer friendly files. This would be useful for the end-client as well the engineers testing the application and can improve productivity.

32. It is important to run the server on a infinite loop as it has to handle multiple clients and accepting and executing the connection from one client would not terminate the server.

33. In java, the end of file is determined by comparing the read output to null when using readLine (text files) and comparing to -1 for binary

files. The file pointer is equal to the sun of Starting point and size of the file.

34. When the back end of an application is designed based on Object oriented Programming with future use in mind, then there would be little to no re-design in the back-end programs and front end can be changed to make the application attractive to the new clients or consumers.

35. Static objects cannot be serialized to file. Serialization is done to an instance of an object and since static objects do not contain an instance, they cannot be serialized.

36. Static methods cannot be overriden because the static objects belongs to a class and is a singleton. It follows that interfaces cannot contain static methods as methods in interfaces are meant to be overriden.

37. The following factors have to be taken into consideration when selecting a data structure to store values: rate of use (rate of create, read, update and delete), size, storage requirements (sorted/unsorted), Technical requirements (memory contiguous or discontiguous, algorithm used).

38. Use of generics/templates is also a type of polymorphism and is called function bounded polymorphism. Also, generics is used to constrain the use of generic classes by inheritance. However, templates takes you away from Object Oriented Programming as templates are multi dimensional and cannot use primitive type in generics.

39. When using Tomcat or other web servers, multi threading need not be implemented explicitly. Tomcat has built in capability to ensure multi threading takes place without data corruption.

40. DefaultSocketClient class is designed so that it extends thread and can handle multiple threads. This is because when the server waits for a connection request from client using the accept() method, the server hangs till it accepts a connection, making other threads to hang. This is avoided by running a separate thread using DefaultSocketClient

class, so that other thread like view can occur when the server thread is waiting.

41. In order to send objects or properties file to the client/server it is important for the object to implement serialization. This is because only serialized objects can be send using object streams.

42. Servlet is written using Java code and contains HTML whereas JSP contains HTML code and contains Java code snippets. JSP does not need to be compiled. Entire JSP page gets translated into a servlet (once), and servlet is what actually gets invoked (for each request).
Using JSP instead of Servlets will avoid writing loads of HTML inside Java println statements and will also avoid maintaintance of HTML code.

43. Post method of forms should be used to send large amount of data and get method to send small amounts of data to the server. Post method is secure than get method because the data sent is not displayed as part of URL header but is actually present within the body.

44.<%! int count = 0 %> JSP declarations can be used to keep count of the number of times the web page is visited. This declaration is called once when the server is first started and not called every time a page loads since it is placed outside the service() method of JSP which is called everytime a servlet/JSP call is made.
<%! int count = 0 %>
<% count++;%>

45. In order to do something which happens only once when a servlet is initially started, include it in the init() method which is called only once during intitial loading the servlet page. It is like a constructor which is called only once.

46.Unlike init() method, service() method is called everytime a request is made to the servlet. When a request is made, a new thread is started and the service() method passes the request to the appropriate methods based on the request (like get, post, delete etc).

47. HTTP is stateless and also connectionless. Connectionless means that after a client makes a request, it disconnects and waits for a response, after the server processes the request, the server reestablishes the connection with the client and send the response. Stateless means that server and client are aware of each other only during the current request. Once the request is fulfilled, it forgets that the later exists. It does not retain information between different requests across web pages.

48. For both doPost and doGet method to do the same things, write the method definitions in one, and call the other. This makes debugging easier.

49. A web application's context path is the directory that contains the web application's WEB-INF directory. It is different from a class path and has to be explicitly passed on to read or write to files.getRealPath() would get the actual path for the servlet's home directory.

```
//reads the data file and stores it in auto object
String path = getServletContext().getRealPath("/")+"/";
auto1 = file1.readData(path + "WEB-INF/Ford_focus.txt", auto1);
```

50. To send an object from servlet to JSP use session.setAttribute(“name”, object) and the jsp page to which the servlet is redirected or posted use session.getAttribute(“name”) to get the object in JSP.

51. Classes can be imported into JSP pages using the <%@ page import =”classname.class” %> tag. This is useful to call variables/methods from the class and also in reusability where frequently used code can be placed in a class and called again and again.

52. To make a java class readable, organize the classes in functionally relevant packages. The first line of the java class would be the package name followed by the imported classes list. Declare the variables in the beginning of the class/methods and initialize it if needed. Also, add comments at every point to indicate what the variable and methods are doing. The variables/method names should indicate the purpose of the variable than using random names. Follow java conventions when naming methods and variables. Use separate

methods for code which would be reused again and again and call the method when needed. Keep the main method as short as possible. Format and indent the code which would also make it eadily readable.

53. Encapsulation is used to hide the objects/variables from other classes in the same package or different packages. Association is used when an object reference has to be passed or returned in the method of another class. Inheritance is used when strong is a relationship and the class is a kind of its parent. Polymorphism is used when a method/class has to implement different roles. For method, overriding or overloading and for classes using parent-child relationship. For example, car is a type of automobile and so is plane/truck.
54. Self-contained and independent classes are those which can function on its own without requiring any other class for it to function or exist. For this prupose, it has to contain all the data variables and method to access/ change the variables in its own class. To design such objects, gather all the required data and write methods for CRUD operations for the required data in the same class.

55. To create a framework, for exposing a complex product, in a simple way and at the same time making your implementation extensible, interfaces have to be used. When a class has to implement some methods and if the relationship to implement method is weak, interfaces are to be used. By using interfaces, functionalities can be passed on to the end user without sharing the actual implementation code. The code implementation can be encapsulated within the internal components of the product or the end user can override the methods to implement their own product definitions. This would make the product extensible and also make the product safe to be exposed.

56. The advantage of exposing methods using different interfaces is again encapsulation. The interfaces can be shared to the end user based on the usage and all the methods need not be available to the end user to use a product.

57. When implementing LinkedHashMap for Auto object in proxyAuto class, a separate class called Fleet was created with a LinkedHashMap of auto objects. An instance of this Fleet class was introduced in proxyAutomobile and methods of Fleet class were called in the methods of proxyAutomobile to change the values. CRUD operations

for the LinkedHashMap are implemented in the Fleet class. This helps in keeping the instances for fleet separate and enables the use of generics which is useful for multi-dimensional programming.