TEMPLATE DESIGN PATTERN

PROGRAM IDEA:

My program is a very simple program that is based on the Template Design Pattern. There are many music shops around the world. Each and every instrument that is kept in such a music store has to be updated in a database. The DB contains basic information such as the stocking number, costPrice, sellingPrice for every instrument. Instead of creating a DB, I am simply printing out these values for every instrument that is added to the store.

WHY THE TEMPLATE DESIGN PATTERN WORKS:

Each instrument in the store has some general information such as stockingNumber (or serial number), cost Price, Selling price. Since all the instruments in the store DB must have these three values associated to them, I can create an abstract class with functions that use these values. Different concrete classes can be created that implement this abstract class, providing different definitions for every instrument. In the homework, the abstract class is InstrumentKind.java and the concrete sub classes are Guitar.java, Drums.java, Bass.java.

FUNCTION OF EACH SUB CLASS:

Each subclass overrides addStockingNumber, addSellingPrice, addCostPrice functions. All these functions do is printout the instrument type and value.

Advantages of using template design pattern:

No code is being duplicated using this method. Since all the concrete classes inherit the super class. Sub Classes can change the implementations of the functions that they are overriding. The flow of the algorithm stays the same, but the implementation of each function in the algorithm differs from class to class.

Disadvantages of using the Template design pattern

It is very easy to overcomplicate the main algorithm with different abstract functions and their varying definitions. If you have a lot of classes, it gets tough to clearly understand the inheritance that takes place.