**Note: In order to run program please go to “consolestore” package and run “FinalRun.java”**

**Console Store Program**

**By: Tan Saeed**

**Overall Program:** This program is a console store simulation. The game starts with an intro showing you the staff, and then creates your store with a budget of 3000$. The game then starts showing you the day and weather. Your first task is to order new system for your store. Then once you do that randomly generated customers will come (some days many and some days little number of customers). Customers will ask for systems and its your job to sell them. The import prices are slightly cheaper than the sell price, so you can eventually profit. This goes on until you choose to close the shop, thus the program ends.

**Inheritance “is a”:** Inheritance is used by several times throughout the program. Console is a super class with HandheldConsole, HomeConsole, and HybridConsole inheriting from the super Console class. The three subclasses also extend to specific systems made by Sony, Nintendo, and Microsoft (used via Abstract Factory design explained later. ConsoleFactory “Is a” abstract super class that has three concrete classes derived from it, that being; Sony, Nintendo, and Microsoft.

**Composition “has a”:** Composition is used by the ConsoleStore class, since it “has a” Account object inside to keep track of money. Another instance is the ConsoleOutlet class that uses the three abstract sub classes of Console, that being HandheldConsole, HomeConsole, and HybridConsole.

**Design Patterns:** The first design pattern was *Singleton*, this was used because we are only supposed to manage one store, so we don’t want to create multiple stores. The ConsoleStore class is only allowed instantiate one store. The other pattern was *Abstract Factory*, this was used because there are multiple types of companies that have different home consoles, this would allow is to grab the right home console depending on the company outlet.

**Polymorphism:** Abstract ConsoleFactory reference is used to instantiate a concrete subclass of Sony, Nintendo, and Microsoft.

**Generics:** The Account class uses generics for its money type ie <T extends Number> making it so only the Number class and its sub classes can be used for T. Although in our case we only use it for Dollars (ie double sense it has decimals) another store might have Yen or Rupees as its currency (an int would make more sense in that case because no decimals).

**Enums:** Used for staff listing in the Job class. This was used to find the name of the staff and his or her job. Albeit in retrospect this does not make much sense for enums class, because enums typically or designed never to change.