**Assignment 6 Design**

**Pseudocode:**

1. **Start**
2. **Interfaces (Already Provided).**
3. **OrderInterface**
4. **BevShopInterface**
5. **Create an enumerated type called DAY. Valid days will be MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY & SUNDAY.**
6. **Create an enumerated type called SIZE. Valid sizes will be SMALL, MEDIUM & LARGE.**
7. **Create an enumerated type called TYPE. Valid types will be COFFEE, SMOOTHIE & ALCOHOL.**
8. **Date Element – Beverage** 
   1. **Create an abstract class called Beverage.**
   2. **Create instance variables for beverage name, beverage type, beverage size, and constant attributes for the base price ($2.0) and size prize ($1 additional to go up).**
   3. **A parametrized constructor to create a Beverage object given its name, type and size.**
   4. **An abstract method called calcPrice that calculates and returns the beverage price.**
   5. **An Overridden toString method: String representation for Beverage including the name and size.**
   6. **An Overridden equals method: checks equality based on name, type, size of the beverage.**
   7. **getters and setters and any other methods that are needed for your design.**
9. **Date Element – subclasses of Beverage**
   1. **Coffee**
   2. **Contains additional instance variables of type boolean to indicate if it contains extra shot of coffee (additional cost of 50 cents) and extra syrup (additional cost of 50 cents).**
   3. **A parametrized constructor**
   4. **An Overridden toString method: String representation of Coffee beverage, including the name, size, whether it contains extra shot, extra syrup and the price of the coffee**
   5. **An Overridden calcPrice method.**
   6. **An Overridden equals method: checks equality based on the Beverage class equals method and additional instance variables for this class.**
   7. **getters and setters and any other methods needed.**
   8. **Smoothie**
   9. **Contains additional instance variables for number of fruits and boolean variable to indicate if protein powder is added to the beverage. The cost of adding protein is $1.50 and each additional fruit costs 50 cents.**
   10. **A parametrized constructor**
   11. **An Overridden toString method: String representation of a Smoothie drink including the name, size, whether or not protein is added, number of fruits and the price**
   12. **An Overridden equals method: checks equality based on the Beverage class equals method and additional instance variables for this class.**
   13. **An Overridden calcPrice method.**
   14. **getters and setters and any other methods needed.**
   15. **Alcohol**
   16. **Contains additional instance variable for weather or not it is offered in the weekend. The additional cost for drinks offered at the weekend is 60 cents.**
   17. **A parametrized constructor**
   18. **An Overridden toString method: String representation of a alcohol drink including the name, size, whether or not beverage is offered in the weekend and the price.**
   19. **An Overridden equals method: checks equality based on the Beverage class equals method and additional instance variables for this class.**
   20. **An Overridden calcPrice method.**
   21. **getters and setters and any other methods needed.**
10. **Data Element – Customer**
    1. **Create a class to represent a customer.**
    2. **Create instance variables for name & age.**
    3. **A parameterized constructor.**
    4. **A copy constructor.**
    5. **An Overridden toString method: String representation for Customer including the name and age.**
    6. **getters and setters and any other methods needed.**
11. **Data Element – Order.**
    1. **Create a class to represent an order, which implements 2 interfaces: OrderInterface & Comparable.**
    2. **Instance variables for order number, order time, order day and customer and a list of beverages within this order**
    3. **A method to generate a random number within the range of 10000 and 90000.**
    4. **A parametrized constructor**
    5. **A method called addNewBeverage that adds a beverage to the order. This is an overloaded method to add different beverages to the order. Refer to the interface OrderInterface provided for you,**
    6. **An Overridden toString method: Includes order number, time, day, customer name, customer age and the list of beverages (with information of the beverage).**
    7. **Override the compareTo method to compare this order with another order based on the order number. Returns 0 if this order number is same as another order's order number, 1 if it is greater than another order's order number, -1 if it smaller than another order's order number.**
    8. **getters and setters and any other methods needed.**
12. **Data Manager – BevShop.**
    1. **Create a class to represent a beverage shop, which implements BevShopInterface.**
    2. **Instance variable for the number of Alcohol drinks ordered for the current order. The current order in process can have at most 3 alcoholic beverages.**
    3. **An instance list to keep track of orders.**
    4. **The minimum age to order alcohol drink is 21.**
    5. **time, order day and customer and a list of beverages Order within this order.**
    6. **An Overridden toString method: The string representation of all the orders and the total monthly sale.**
13. **End**