**Coffee Pot Data Dictionary**

***Format:***

CLASS: class description

**(if not void)TYPE:METHOD(***Param type***):** method description

*Attribute:* attribute description, includes default value if it has one.

Beverage: Implements default methods for the beverages only.

* **dispense():** dispenses the product
* **print():** print out the type of product
* **isAcceptableCondiment(***String***):** checks to see if the condiment is an “acceptable” condiment for the designated beverage.
* **Beverage():** constructor
* *String name:* name of the beverage.
* *String description:* what the beverage is.
* *int inventory:* how much of some item is left in the machine.
* *int price:* how much the Beverage costs.

Change Machine: keeps track of any monetary interaction and inventory by amounts and totals, manages changeDue and determines if paying with rCard or cash.

* **changeMachine():** Constructor
* **selectPaymentType():**  Determine if paying with cash of rCard.
* **checkout():** deducts credit from rCard or from moneyInserted to determine change due or balance left.
* **coinReturn():** returns the change of moneyInserted.
* **hasPaidEnough():** Has the person put in enough money to pay for the Beverage.
* **addToPurchase(***int***):** add the item’s price to the purchase amount.
* **printReport():** prints a report of the machine’s “money” by specific counts of coins and money made in cash or rCard or both.
* **makeChange(***int***):** calculate the change and whether we which type of coins we are returning.
* **insertNickel():** inserts 5c to the machine.
* **insertDime():** inserts 10c to the machine.
* **insertQuarter():** inserts 25c to the machine
* **insertDollar():** insert 100c to the machine
* **returnNickel(***int***):** returns a nickel and deducts 5c from change due and reduces the nickel count by 1
* **returnDime(***int***):** returns a dime and deducts 10c from change due and reduces the dime count by 1
* **returnQuarters(***int***):** returns a quarter and deducts 25c from change due and reduces the quarter count by 1
* **returnDollars(***int***):** returns a dollar and deducts 100c from change due and reduces the dollar count by 1
* *int nickels:* keeps a count of the nickels inside the machine.
* *int dimes:* keeps a count of the dimes inside the machine.
* *int quarters:* keeps a count of the quarters inside the machine.
* *int dollars:* keeps a count of the dollars inside the machine.
* *int cashBalance:* how much has been inserted cash.
* *int moneyInserted:* total of how much has been inserted.
* *int rCardBalance:* how much is available from the rCard.
* *int totalPurchase:* amount to pay for the whole order.
* *boolean payWithRCard:* when true goes through code to pay with rCard.

Coffee: a Beverage option

* **dispense():** dispenses coffee, and reduces coffee inventory by 1.
* **hasEnough(***int***):** checks if there is enough coffee in the inventory to complete the order.
* **isAcceptableCondiment(***String***):** adds acceptable condiments to the Coffee beverage.
* *String name:* “Coffee”
* *String description:* “Black Coffee”
* *int inventory:* how much coffee there is in the inventory, 100.
* *int PRICE:* NON-NEGOTIABLE price for a coffee, 35c.
* *String[] condiments:* The condiments that coffee can have, “Sugar” and “Cream”.

CoffeePot: Builds and prints the menu, manages the output.

* **CoffeePot(***Beverage***):** Builds the menu.
* **printMenu():** prints the menu.
* *BeverageComponent allMenus:* attribute needed as a temp to build the menu for each new item.

CoffeePotGUI: The graphical interface for the user to input the order.

* **CoffeePotGUI2():** constructor
* *int creamWanted:* how much cream is wanted.
* *int lemonWanted:* how much lemons is wanted.
* *int sugarWanted:* how much sugar is wanted.

CoffeePotMain: class dedicated to running the program.

* **Main(***String[] args***):** runs the whole program

Order: Maintains a list of Beverage components in the current order.

* **Order():** constructor
* **add(***BeverageComponent***):** adds a beverage component to the order.
* **remove(***BeverageComponent***):** removes a beverage component from the order.
* **cancelOrder():** cancels the entire order and sets all elements to default.
* **dispense():** finalized and dispenses an order.
* **printOrder():** outputs the current order to the console.
* *ArrayList<BeverageComponent> order:* array list to create an order.

BeverageComponent: Implements default methods for beverage and condiment classes.

* **add(***BeverageComponent***):** adds a new component to the menu.
* **boolean: isAcceptableCondiment(***BeverageComponent***):** checks if the condiment is acceptable to the Beverage.
* **dispense():** dispenses the beverage and adjusts the inventory.
* **print():** prints the dispense of a Beverage.

Condiment: Implements default methods for the condiments only.

* **print():** prints the condiment added to a beverage and the amount of it.

Cream: Condiment to add to either Coffee or Tea

* **dispense():** dispenses condiment and adjusts inventory.
* *String name:* “Cream”
* *String description:* “Non-Dairy Creamer”
* *int inventory:* keeps track of how much “cream” we have left, 100.

Decaf: a Beverage option

* **dispense():** dispenses beverage and adjusts inventory.
* **boolean: isAcceptableCondiment(***String***):** adds acceptable condiments to the beverage.
* *String name:* “Decaf”
* *String description:* “Decaf Coffee”
* *int inventory:* keeps track of how much “decaf” is left, 100.
* *int PRICE:* NON-NEGOTIABLE price, 35c.
* *String[] condiments:* The condiments that decaf can have, “Cream” and “Sugar”.

HotChocolate: a Beverage option

* **dispense():** dispenses beverage and adjusts inventory.
* **boolean: isAcceptableCondiment(***String***):** adds appropriate condiments to the beverage.
* *String name:* “Hot Cocoa”
* *String description:*“Hot Chocolate”
* *int inventory:*keeps track of how much hot chocolate we have left, 100.
* *int PRICE:*NON-NEGOTIABLE price, 75c.
* *String[] condiments:*the condiments that hot chocolate can have, “Marshmallows”.

Lemon: Condiment to add to Tea.

* **dispense():** dispenses condiment and adjusts inventory.
* *String name:* “Lemon”
* *String description:* “Lemon Concentrate”
* *int Inventory:* keeps track of how much “lemon” is left, 100.

Marshmallow: Condiment to add to Hot Chocolate.

* **dispense():** dispenses condiment and adjusts inventory.
* *String name:* “Marshmallow”
* *String description:* “

Menu: A literal menu for the user to look at his options in order to decide what to order.

* **menu(***String, String***):** constructor
* **add(***BeverageComponent***):** adds a new component to the menu.
* **remove(***BeverageComponent***):** removes a component from the menu.
* **print():** prints the menu for the user to ponder upon.
* *ArrayList<BeverageComponent> menuComponents:* arrayList to build the menu.
* *String name:* name of the new item to add to the menu.
* *String description:* add a description to an item on the menu.

ProductList: Builder for the menu, manages what goes in and out of the menu.

* **ProductList():**  constructor
* **allCondiments():** adds all the condiments to an array list that then builds the menu.
* **allBeverages():** adds all the beverages to an array list that then builds the menu.
* *Condiment cream, sugar, lemon, marshmallow:* all the condiment objects needed to access their methods.
* *Beverage coffee, decaf, hotCoco, tea, soup:* all the beverage objects needed to access their methods.
* *BeverageComponent coffeeCondiments:* the Menu object to add and build.

Soup: a Beverage option

* **dispense():** dispenses beverage and adjusts inventory.
* **boolean: isAcceptableCondiment(***String***):** adds appropriate condiments to the beverage.
* *String name:* “Soup”
* *String description:* “Chicken Noodle”
* *int inventory:* keeps track of how much soup is left, 100.
* *int PRICE:* NON-NEGOTIABLE price, 75c.
* *String[] condiments:* acceptable condiments for “soup”, none.

Sugar: Condiment to add to Coffee or Tea.

* **dispense():** dispenses condiments and adjusts inventory.
* *String name:* “Sugar”
* *String description:* “All-natural Sugar”
* *int inventory:* keeps track of how much “sugar” we have left, 3.

Tea: a Beverage option

* **dispense():** dispenses beverage and adjusts inventory.
* **boolean: isAcceptableCondiment(***String***):** adds appropriate condiments to the beverage, Tea.
* *String name:* “Tea”
* *String description:* “Black Tea”
* *int inventory:* keeps track of how much “tea” we have left, 100.
* *int PRICE:* NON-NEGOTIABLE price, 60c.
* *String[] condiments:* Acceptable condiments that “tea” can have, “Cream”, “Sugar”, and “Lemon”.