

## Lab 7: Expanding a Coffee shop software

In this lab, you will be implementing the Decorator design pattern.

### Part I: Adding New Blends

Add two new blends: **Hazelnut** and **Caramel** in addition to the four given blends (DarkRoast, Decaf, Espresso, and HouseBlend). The Hazelnut blend should cost 1.25 and the Caramel blend should cost 1.35. Remember to extend the appropriate class, or to implement the appropriate interface.

- Hazelnut class
  - constructor
  - cost() method - returns the cost of the blend.
- Caramel classes
  - constructor
  - cost() method - returns the cost of the blend.

### Part II: Adding New Condiments

Add three new condiments: **Cinnamon**, **Mint**, and **Chocolate** in addition to the four given condiments. Cinnamon and mint each add 15 cents to the cost of the drink and Chocolate adds 20 cents. All condiments should add their name to the description. Again, remember to extend the appropriate class, or to implement the appropriate interface.

- Cinnamon class
  - getDescription() method - returns the string formed by concatenating beverage description with the condiment name.
  - cost() method - returns the sum of the cost of the condiment and the cost of the beverage.
- Mint class - same as above class
- Chocolate class - same as above class

### Part III: Driver program (StarbuzzCoffee.java)

Modify the driver class, StarbuzzCoffee, that allows a user to order each of the new blends. Your driver should also demonstrate that a user can add each of the four new condiments that dynamically changes the total price of the ordered coffee depending on the chosen blend and condiments. Be sure to do the following:

- Order 1 of each type of beverage you created.

- Add multiple toppings to each ordered beverage.
- Use each of the new condiments at least once.

Display the description of the beverages you created along with their cost.

**Program compilation & execution:**

```
> javac *.java
```

```
> java StarbuzzCoffee
```