**Lab 7**

|  |  |  |  |
| --- | --- | --- | --- |
| Student Name | | Student CSUSM ID | Contribution percentage |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**Grading Rubrics (for instructor only):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | 1. Beginning | 2. Developing | 3. Proficient | 4. Exemplary |
| Modeling | 0-14 | 15-19 | 20-24 | 25-30 |
|  |  |  |  |
| Program: functionality  *correctness* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Program: functionality  *Behavior Testing* | 0-9 | 10-14 | 15-19 | 20 |
|  |  |  |  |
| Program: quality ->  *Readability* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Modularity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Program: quality ->  *Simplicity* | 0-2 | 3-5 | 6-9 | 10 |
|  |  |  |  |
| Total Grade (100) |  | | | |

**Problems:**

Develop a versatile coffee making machine (CMM). The CMM has a number of functionalities: setCoffeeType(String str), setGrindingTime(int secs), addCondiment(CondimentIF type), setTemperature(int degree), holdTemperature(int seconds), wait(int seconds), setPowerLED (int num), setTypeLED (int num), computePrice(CoffeeIF cif), and done( ) to finish running.

When the CMM is powered on, its menu allows a user to select one from a list of 5 built-in programs that make Regular, Mocha, Latte, Espresso, and Cappuccino coffees, respectively. All programs have a start() method to start its operation. Due to memory constraints, at any time only one program can be loaded into the run-time memory from the external flash memory. The CMM has an LED display with two digits. The first digit is either 0 running or 1 not running. The second digit is from 1 to 5, indicating the currently chosen program.

The menu prices for coffees without condiments are $1/cup for regular, $2/cup for Mocha, $3/cup for Latte, $4 for Espresso, and $5/cup for Cappuccino. The machine offers a variety of condiments. The charge each serve is $.25 for cream, $.50 for vanilla, and $1 for chocolate. All condiments can be mixed and added repeatedly as desired and the sale price will simply add up. The CMM stores their sale records into a data structure List<CoffeeIF>.

The start() operation of the Regular program run the followings in order.

1. Indicate the machine as running and display the purchased type of coffee to LED.
2. Grind coffee beans for 8 seconds.
3. Heat up the water to 150 degree F and hold the temperature for 2 seconds.
4. After 15 seconds, set the LED to indicate the machine is not running.

The start() operation of the Mocha program run the followings in sequence.

1. Indicate the machine as running and display the purchased type of coffee to LED.
2. Grind coffee beans for 5 seconds.
3. Heat up the water to 200 degree F and hold the temperature for 5 seconds.
4. Add cream and vanilla once.
5. After 15 seconds, set the LED to indicate the machine is not running.

You may implement the start() operation of the other programs at your own choice.

Let different coffee types and condiments have a menu price field. However, there is no variable to store the final sale price for each cup of coffee sold. The final sale price must be computed through the coffee machine’s method computePrice(CoffeeIF cif). For simplicity, the other methods only require a message prompt. For example, setGrindingTime(2) displays a message like “Grinding for 2 seconds.” The return types of the methods will be your own discretion. The done() method completes the transaction of a cup and adds the sold coffee object to the List.

Please prepare a main program to test your software. There will be two cups of coffee sold. Show the necessary operations of the machine and finally print out the sale price for each.

1. A cup of Mocha with one serve of chocolate added.
2. Besides the cream and vanilla, a cup of Espresso with also two serves of chocolate added.

**Solution:**

* First, remember to zip the src folder of your project and submit the zip file to the ungraded assignment named “Lab7CodeSubmission”. One submission from each team.
* Paste your design in UML class diagram here.
* Paste a screenshot of a run of your program here.
* Also paste all you source code here.
* Save this report in PDF, then submit the pdf report to the graded assignment named “Lab7ReportSubmission”. One submission from each team