

Project #4 (55 points)

Due Date

Tuesday, April 6, by 11:59pm.

Submission

- You must include both team members' names in the comment block on top of EVERY Java file.
- Your project folder must include the following **subfolders/files for grading**.
 - Source folder [50 points]
 - All the .java files
 - All the .fxml files
 - Javadoc folder [5 points]

Project Description

RU Café sells donuts and coffee. Your team will develop a software for the Café to manage the orders. Your team must use JavaFX to develop the GUIs for taking the orders, placing the orders and cancelling an order. The user can take one order at a time. The menu items include

- 3 types of donuts: yeast donut, cake donut and donut holes. Regardless of the flavors, a yeast donut is \$1.39, a cake donut is \$1.59, and a donut hole is \$0.33.
- Brewed coffee, with the choices of different add-ins: cream, syrup, milk, caramel, and whipped cream; and, with the choices of different sizes: Short, Tall, Grande and Venti. The price for a Short black coffee is \$1.99. Each add-in costs \$0.20. The price increase for the next size is \$0.50. For example, the price increase for a Grande is \$1.00, therefore a Grande black coffee is \$2.99, and a Grande coffee with cream and syrup would be \$3.39.

Your software must provide 5 GUIs below. **-5 points** for each GUI missing. The user can navigate between the GUIs to add/remove menu items to/from an order, review/place the order, check all orders and cancel an order.

- (1) Main Menu, which shall include the options of ordering donuts, ordering coffee, show the current order, and show all the orders placed by the user.
- (2) Ordering Donuts, which allows the users to add or remove the donuts to the order. The GUI shall provide the options of 3 donut types: yeast donut, cake donuts and donut holes. Each donut type provides at least 3 different donut flavors. Upon selection of the donut type and the specific donut flavor, the user can also select the quantity of the donut selected. While adding or removing the donuts, the GUI shall display the sub-total of the selected donuts dynamically, with 2 decimal places.
- (3) Ordering Coffee, which allows the user to add the coffee to the order. The GUI shall provide the options of the sizes: Short, Tall, Grande and Venti, and the options of the add-ins: cream, syrup, milk, caramel, and whipped cream. While adding or removing the add-ins or changing the sizes, the GUI shall display the sub-total of the coffee dynamically, with 2 decimal places.
- (4) Current order detail, which includes all the menu items added to the order. Each item in the order shall include the details of the item name, quantity, add-ins (for coffee only) and size (for coffee only.) The user can review the order, remove a selected item and place the order. The GUI shall display the sub-total, sales tax and total amount of the order dynamically. The New Jersey Sales and Use Tax Rate is 6.625%. All dollar amounts must be displayed with 2 decimal places.
- (5) Store orders page, which lists all the orders placed by the user. The GUI shall list the detail of each order and display the total amount of each order, with 2 decimal places. The user can select an order and cancel the order. The GUI shall display the remaining orders after the cancellation. The user can also save/export the store orders to a text file, which shall include the details of every order, consistent with the details displayed on the GUI.

Project Requirements

1. This is a **group assignment**. You must work with your designated partner in order to get the credit for this project.
2. You **MUST** follow the software development ground rules, or **you will lose points** for not having a good programming style.
3. Each Java class must go in a separate file. **-2 points** if you put more than one Java class into a file.
4. For each GUI, you must create a .fxml and its associated controller class. That is, you will have 5 .fxml files and 5 controller.java files. **-5 points** if this is not done properly.
5. When the user is navigating between the GUIs, the primary stage (main menu) must remain visible, **or -5 points**.
6. You can use any JavaFX UI controls to design your GUIs. However, you **MUST** include the following JavaFX UI controls, or **-3 points** for each violation.
 - (a) ComboBox
 - (b) ImageView
 - (c) ListView
 - (d) CheckBox
7. **You are NOT ALLOWED to use System.out in ALL CLASSES, or you will lose 10 points.** Use either a TextArea or Alert pop-ups to display any messages.
8. You can use any Java library classes. However, you must include the following classes. **-5 points** for each class missing OR not used.
 - (a) include the following Java interface to your project

```
public interface Customizable {
    boolean add(Object obj);
    boolean remove(Object obj);
}
```
 - (b) **MenuItem class**. This is the superclass of all menu items, such as donuts and coffee. Any class defined for a menu item must extend this class. All the subclasses must include a “**itemPrice**” method for calculating the price of the menu item. **-2 points** for each violation.
 - (c) **Coffee class**. An instance of this class is a menu item in an order. This class must extend MenuItem class and implement the Customizable interface above, or **-2 points**.
 - (d) **Order class**. An instance of this class has a unique order number and keeps the list of menu items added by the user. This class must implement the Customizable interface above, or **-2 points**.
 - (e) **StoreOrders class**. An instance of this class keeps the list of orders placed by the user. This class must implement the Customizable interface above, or **-2 points**.
9. You are required to **generate the Javadoc** after you properly commented your code. Your Javadoc must include the documentations for the constructors, private methods and public methods of all Java classes (*.java files.) You **must comment the Main.java and All controller classes** and include them in the Javadoc. **DO NOT** include the *.fxml file, which is NOT a java file. Generate the Javadoc in a single folder and include it in your project folder to be submitted to Canvas. You are responsible to double check your Javadoc after you generated them. The grader will navigate the Javadoc with the “index.html”. You will **lose 5 points** for not including the Javadoc, OR, the grader cannot navigate your Javadoc through the “index.html”.

System Testing

1. You are responsible to thoroughly test your software and ensure your software is meeting the requirements listed above. You will **lose 2 points** for each incorrect implementation, incorrect amount or incorrect information shown on the GUIs.
2. Your software must always run in a sane state and **should not crash in any circumstances**. You must catch all Java Exceptions. Your program shall continue to run until the user stops the program execution or closes the window. **You will lose 2 points** for each exception not caught.