Julissa Mota  
Professor Kasey Nguyen  
CIS-18A  
30 July 2020  
Company Name: Pro-J

Final Project: Documentation

The Codecakes program, designed by Pro-J, is created specifically for the Codecakes Company, a cupcake store that allows their customers to choose from different flavors for each of their cupcake’s tier: the base, frosting, and topping. This program is available for customers to use, provided on their cell phone as an application from the company, and also for the staff operation which is provided at their store POS systems. Our company’s purpose to creating a program for Codecakes would be to guide the customer and staff operations through a step-by-step decision menu which displays the flavors available for each tier, one tier at a time. The program begins with a welcome message, reminding the user where the customer is ordering from. The first step has begun which asks the user to pick the flavors that are provided with number choices. When the number choice has been input, the program will reassure the customer that the choice was correct by displaying again what they have chosen. After, the second step will display for frosting, asking the user to, again, pick from flavors provided with number choices, displaying what they have picked, and repeats this with step three’s topping choice. The menu provides a “quit” option for the user to cancel their order before the order it is processed after the final decision. After the third step, the program will process the customer’s order and display a detailed receipt that lists what the customer has chosen for each tier alongside their price, ultimately providing a receipt with tax. Before the program ends, it provides a “thank you” message and instructions for payment. Our Codecakes program will ensure that a customer has their satisfaction with their experience alongside their order. This program is a simple design, customized for easy and smooth use, preventing confusion and complications. Our goal is to provide a satisfied experience for the customers and staff for Codecakes, as customers will enjoy their delicious cupcakes that they have customized through our menu, and the staff having easy and quick processes for their customers. It is quick, easy, and ultimately delicious.

Final Project: Pseudocode

Import package “codecakespack” that Codecakes is included in.  
Import java.util.\* to the program.  
Include interface “opening” that includes a void message “msg1” for the beginning of the program. Include interface “processing” that includes a void message “msg2” for the processing message in the program.  
Class WelcomeMenu that implements opening, processing, and includes interface msg1 and msg2 outputs.  
Class UserChoice that contains integers base\_choice, frosting\_choice, and topping\_choice for the user input choices.  
Main class Codecakes { , throws java.io.IOException (for input)  
Construct objects for UserChoice, being userpick, Codecakesendpkacage.Closure, being bye, and for WelcomeMenu, being ob1.  
Call/display interface opening, msg1 (outputs the introduction to the program).  
One-dimensional array for three set prices; one for each tier.  
Variables: int i (for switch statement), double base\_total = 0, double frosting\_total = 0, double topping\_total = 0, double receipt = 0 (each for adding the prices as the program runs through each menu of each step).  
Output for Step 1, menu displays for base flavors using a switch statement.  
Input gather from user through userpick.base\_choice; must be between choices 1-4 (4 being quit program).  
IF statement for choices 1-4. 1-3 stores choice for flavor, 4 quits program, anything above or less than choices 1-4 is invalid and exits program. Choices 1-3 allows base\_total to store/add array prices[0].  
Output for Step 2, menu displays for frosting flavors using switch statements.  
Input gather from user through userpick.frosting\_choice; must be between choices 1-4 (4 being quit program).  
IF statement for choices 1-4. 1-3 stores choice for flavor, 4 quits program, anything above or less than choices 1-4 is invalid and exits program. Choices 1-3 allows frosting\_total to store/add base\_total + array prices[1].  
Output for Step 3, menu displays for frosting flavors using switch statements.  
Input gather from user through userpick.topping\_choice; must be between choices 1-4 (4 being quit program).  
IF statement for choices 1-4. 1-3 stores choice for flavor, 4 quits program, anything above or less than choices 1-4 is invalid and exits program. Choices 1-3 allows topping\_total to store/add frosting\_total + array prices [2].  
Add receipt\_total + topping\_total and store into receipt\_total to hold overall addition value  
Call/display interface processing, msg2 (ouputs that the order is processing and is setup for receipt menu).  
IF userpick.base\_choice is equal to 1-3, displays choices and prices from array prices[0].  
IF userpick.frosting\_choice is equal to 1-3, displays choices and prices from array prices[1].  
IF userpick.topping\_choice is equal to 1-3, displays choices and prices from array prices[2].  
Outputs reciept\_total (which stored the addition of the prices).  
Multiply receipt\_total \* tax\_rate to store into only\_tax (to calculate only tax price).  
Add receipt\_total + only\_tax to store into tax\_total (to calculate the total plus tax).  
Construct objects for BigDecimal, being bd, to convert only\_tax to scale in 2 place decimal form and store into only\_tax\_dec.  
Construct object for BigDecimal, being bd1, for tax\_total to scale in 2 place decimal form and store into tax\_total\_dec.  
Display only\_tax\_dec and tax\_total\_dec.  
Call bye.msg(), which is from the Closure class in the package: Codecakespack.  
Program is ended.