Pseudocode

* Create package that contains the items of the menu
* Separate the items into classes
  + CLASS coffee
    - Coffee options as instance variables (3 options)
    - Constructor coffee() initializes the variables of data type int
    - VOID method FirstMenu() displays the variables in the menu with their order number
  + CLASS other drink options
    - Drink options as instance variables (4 options)
    - Constructor other() initializes the variables of data type int
    - VOID method SecondMenu() displays these variables in the menu with their order number
  + CLASS pastries
    - Pastry options as instance variables (3 options)
    - Constructor pastries() initializes the variables of data type int
    - VOID method ThirdMenu() displays the pastry options in the menu with their order number
* End of package items
* Create a new package called menu
* Import package JAVA.UTIL.\* for the scanner
* Import package items to access all classes in that package
* Create an interface containing an array of prices
  + DEFAULT method cost() holds the array
    - Price array holding the prices of the items
* Create an interface with abstract method
  + Abstract method for checkout()
* Create an outer class that holds the execution of main()
  + CLASS CoffeeShop displays the menu
    - Include clause to read input
    - Declare objects for the imported package (3 objects)
    - Declare variable for the count of data type int
    - Declare variables for the total, choice, sum, tax, SalesTax of data type int
    - Output welcoming line “Welcome to Cozee Coffee!”
    - Display “Menu” newline
    - Display on newline: “0-Complete order and checkout”
    - Create reference to imported classes menu with their respective object
* Create an inner class that implements the two interfaces
  + - CLASS order takes the order of the customer and displays the receipt
      * Use interface method checkout()
        + Declare scanner
        + Create Boolean variable for input validation
        + Use the Do-While loop for repetition of the input validation

Display “How many items will you order today?”

Create if else condition to check if input is an integer

IF is an integer

Variable Count = input scanner

Boolean value is true

ELSE

Display “Invalid. Enter an integer.”

Boolean value is false

Clear input buffer

* + - * + While condition is (!(boolean))
        + Use a FOR loop controlled by the count variable

Display “Enter your order number at a time”

Variable choice = input

Use the WHILE loop for input validation

Choice <0 or choice >10

Display “That is not a valid order number”

Choice =input

Use the SWITCH once a valid choice is inputted

Create cases 0-10

Cases correspond to the order number choice

Display “You ordered (choice)”

Display “that will cost” price array [choice]

BREAK

CASE 0

Display “End of order”

BREAK

Add the prices

Sum += array[choice]

* + - * + End of the FOR loop
      * Display “subtotal” show sum
      * Intialize tax = 0.0775
      * Calculate sales tax = tax \* sum
      * Display total after tax.
      * Display thank you message

Project Documentation

For this coding project, I have created a program for a small coffee shop. My coffee shop is called Cozee Coffee. This program is designed for the use of customers. They will be able to pay virtually without person to person contact. This will minimize waiting lines, an overcrowding shop, and reduce the workload on baristas. The program is designed to display the menu and the customers can choose any of the available options. The menu includes a variety of coffee types, tea, hot chocolate, cold drinks, and pastries. My coffee shop is not only for coffee lovers, but also anyone who is craving a hot or cold drink along with a sweet desert. This offers a new way to implement social distancing. The baristas will no longer be taking direct orders, instead they will only be behind the counter making the customer’s order. Orders will be taken virtually. The programs technical functions consist of displaying the coffee shop’s menu along with the prices. Each item is associated with its adequate price. Within the program, this is stored in an array. First, they will input the number of items they will order. Then they will input their order. Their input will be checked or validated. Each item will be saved into the program until they state the end of their order. Once the program has all the items, the program will add all the prices. Finally, it will display their order and total amount. It also displays a thank you message, and lets the customer know their order will be done shortly. My program is separated into classes each with its specialized function. This will simplify the code for any programmer who may need to adjust the program in the future or if technical difficulties arise. My program implements the programming concepts in Java; therefore, the programmer must be knowledgeable in the Java programming language. Cozee Coffee deeply cares about the safety and concern of their customers and staff members. This technological advancement will prevent the spread of germs by minimizing human contact. This is essentially important during this time of COVID-19.