

Project 1 Pseudocode

Function Name: CalculateBonusBucks

Parameters: prevAmountSpent, totalPrice

Return: (int) bonusBucks

- If the previous amount spent is greater than or equal to 200
 - Calculate Bonus Bucks
 - $((\text{prevAmountSpent} + \text{totalPrice}) - (\text{prevAmountSpent} - (\text{prevAmountSpent} \% 5))) / 5$
 - Return bonusBucks
- Else
 - Return -1

Function Name: PrintOutput

Parameters: regularCustomerArray, preferredCustomerArray

Return: void

- Create a printwriter for the two output files.
- For 0 to regularCustomerArray.length()
 - Print data in regularCustomerArray to “customer.dat”
- For 0 to preferredCustomerArray.length()
 - Print data in preferredCustomerArray to “preferred.dat”

Function Name: TotalPrice

Parameters: drinkType, drinkSize, totalQuantity, costPerSqInch

Return: (double) totalPrice

- Calculate the total price to design the total drinks wanted by the user
 - If size = ‘S’
 - Price of designing = $(2.0 * \text{PI} * (\text{Diameter} / 2.0) * \text{Height} * \text{costPerSqInch})$
 - If drink is “soda”
 - Price of drink = $(\text{sodaPrice} * \text{smallDrinkOunces})$
 - Else If drink is “tea”
 - Price of drink = $(\text{teaPrice} * \text{smallDrinkOunces})$
 - Else

- Price of drink = (punchPrice * smallDrinkOunces)
 - Else If size = 'M'
 - Price of designing = $(2.0 * \text{PI} * (\text{Diameter} / 2.0) * \text{Height} * \text{costPerSqInch})$
 - If drink is "soda"
 - Price of drink = (sodaPrice * mediumDrinkOunces)
 - Else If drink is "tea"
 - Price of drink = (teaPrice * mediumDrinkOunces)
 - Else
 - Price of drink = (punchPrice * mediumDrinkOunces)
 - Else If size = 'L'
 - Price of designing = $(2.0 * \text{PI} * (\text{Diameter} / 2.0) * \text{Height} * \text{costPerSqInch})$
 - If drink is "soda"
 - Price of drink = (sodaPrice * largeDrinkOunces)
 - Else If drink is "tea"
 - Price of drink = (teaPrice * largeDrinkOunces)
 - Else
 - Price of drink = (punchPrice * largeDrinkOunces)
- Return ((Price of Designing + Price of drink) * totalQuantity)

Function Name: readPreferredCustomer

Parameters: preferredCustomerFileName

Returns: preferredCustomerArray

- While not end of file
 - Read a line and increment number of customers by 1
 - Close file
- Create a preferredCustomerArray with the number of customers
- While not end of file
 - Read the line with data about customer
 - Customer id, first name, last name, amountSpent, and discount/bonusbucks
 - Input information from line into the designated variables in the classes
 - Cast amount spent and discount/bonus bucks into a double
 - Create a new preferredCustomer object and set values to it.

- Add it to preferredCustomerArray[i]
- Return preferredCustomerArray

Function Name: readRegularCustomer

Parameters: regularCustomerFileName

Returns: regularCustomerArray

- While not end of file
 - Read a line and increment number of customers by 1
 - Close file
- Create a regularCustomerArray with the number of customers
- While not end of file
 - Read a line and split it into 4 attributes by spaces
 - Customer id, first name, last name, and amount Spent
 - Cast amount spent into a double
 - Create a new Customer object and set values to it.
 - Add it to regularCustomerArray[i]
- Return regularCustomerArray

Function Name: StatusUpdate

Parameters: prevAmountSpent, discountedAmount, regularCustomerArr, preferredCustomerArr

Return: boolean

- totalAmountSpent = (prevAmountSpent + discountedAmount)
- If totalAmountSpent >= 200
 - Change the customer object from Gold to Platinum
 - Update bonus bucks instead of discount
 - Return true
- Else if totalAmountSpent >= 50
 - Resize the preferredCustomerArr by increasing size by 1
 - Add the customer information to the end of the preferredCustomerArr
 - Remove customer information from regularCustomerArr
 - Reduce regularCustomerArr size by 1
 - Return true
- Else
 - Return false

Function Name: ApplyDiscount

Parameters: prevAmountSpent, currentTotal, customerDiscount

Return: (double) newTotal

- If prevAmountSpent is between 50 and 200
 - Calculate the new amount after the discount is applied
 - $\text{newTotal} = (\text{currentTotal} - (\text{currentTotal} * \text{customerDiscount}))$
 - Return newTotal
- Else
 - Return -1.0

Function Name: Main

- Prompt the user to enter the file names
- Ask the user to enter the **regular customer, preferred customer and orders file names** in the specified order.
- Enter the information from the regular customer file into the regularCustomerDataArray[]
- Check to see if the preferred customer data file is empty
 - If file is not empty
 - Create the array and input the information from the file into the preferredCustomerDataArray[]
- Loop until the end of the orders file
 - Validate the input from the orders file
 - If valid
 - Calculate the totalAmount of purchased items
 - Process the order
 - If eligible for promotion
 - If first preferredCustomer
 - Create preferredCustomerArray
 - Promote
 - Else
 - Promote
 - Collect next line with data from orders file
 - Else
 - Loop back and get the next line with data from orders file
- Print final data to output files