Project 2 – psuedocode

Diem Pham

NET id: dtp160130

CS2336.002

Classes:

• Customer (base class)

o Members

First name

Last name

Guest ID

Amount spent

o Methods

Overloaded Constructor

Accessors

Mutators

• Preferred Customer (derived class)

o Member

Discount Percentage

o Methods

Overloaded Constructor (chained w/ base class constructors)

♣ Accessor

♣ Mutator

getDrinkPrice(( type of drink, number of ounces){

double price = pricePerOunce \* numberOfOunces

return price

}

getDrinkSize(radius, height){

double size = 4\*pi\*radius\*height

return size;

}

getGraphicPrice(price per square inch){

double price = DrinkSize \* price per square inch

return price

}

amountSpent(quantity){

double total = (getDrinkPrice + getGraphicPrice) \* quantity

return amountSpent

}

double updateTotalCost(){

call amountSpent method

accumulate amount spent to totalCost

return totalCost

}

Read customer information and return it into array (customer information) throws IOException{

Return array

}

Boolean isPreferredstatus(){

Call updateTotalCost method

If the updateTotalCost has accumulated $150 or more

Return status = true

Else return status = false

}

Boolean doesCustomerExist(customer id, type of customer){

Return true if customer exists in a specific type of customer

Else false

}

Resize the preferred customer array(Customer customer){

Add the new preferred customer into the end of the preferred customer array

Return preferred customer array

}

Resize the regular customer array(Customer customer){

Push all the row forward after the promoted customer taken

Return array

}

Double Discount(totalCost){

Double discount;

If spent < $150

Double = 0.0

If spent $150

Double = 0.05

If spent $200

Double = 0.07

If spent $350

Double = 0.1;

Return discount

}

Main(){

//read preferred customer information, regular customer information, and order stored in files

Read customer information and return it into array(preferred customer) throws IOException{

Return arrayOfPreferredCustomer

}

Read customer information and return it into array(regular customer) throws IOException{

Return arrayOfRegularCustomer

}

Read customer information and return it into array(order) throws IOException{

Return arrayOrder

}

//Processing an order, call those methods as below to process the order

double getDrinkPrice( type of drink, number of ounces)

double getDrinkSize(radius, height)

double getGraphicPrice(price per square inch)

double amountSpent(quantity)

double updateTotalCost()

// check if the customer in regular or preferred list

If in preferred

//See how much discount the customer get

If in regular

//See if the customer is qualified to get promoted to preferred list

If (isPreferredStatus()){

Resize regular customer array()

Resize preferred customer array()

}

Discount(totalCost)

haveToPay = amountSpent() \* Discount()

In the end, write data to respective files

}