Class:

PepperoniHut extends Order()

Method name (input type): return type

Public static void main (String args[]) {
//make instances of subclass objects for nav
Order orderticket = new Order();
Replenish replenishTicket = new Replenish();
Checkout check = new Checkout();

//create table for info storage int table [][] = new int [10] [5];

//switch loop for main menu selection Do{ } (while userOp < 4);

//display menu options

71-----

Kitchen extends Order

Member variable: type

Int cheese; int sausage int dressing int steak int coke int sauce; int crust int buns int bread int pepsi int pepperoni int lettuce int chicken int garlic int rootB int mushroom int crouton int fish int butter int sprite

Method name (input type): return type

Public void orderLine(int pizza, int salad...) {
//call makeFood methods that execute object creation

//create food item to be delivered to customer Kitchen foodToEat = new Kitchen(); foodToEat.name = "food"; LowAlert();

//alerts employees when ingredients quantities are low Public void LowAlert () {
Int ingredients [] = {list of all ingredients}
Int min = 2;
String itemName [] = {list of all item names in order}
For (int i = 0; i < 20; i++) {
If (ingredients[] <= min) {

Class:

Checkout extends Replenish

Member variable: type

Sout running low on + itemName [i]

Double tax = 0.7 double garlicBreadPrice = 5.99
Double pizzaprice = 12.99 double fountainDrinkPrice = 1.99
Saladprice = 9.99 double toppingPrice = 1.25
sanwich Price = 6.99

Method name (input type): return type

Public void checkout (int table [][]) {
//calls upon table order & displays receipt
//prompts for table number
//calculates food totals with stacked if blocks
If (table [tableNum] [#] > 0) {
foodTotal = (int) (table [tableNum] [0] * foodPrice);

//calculates total price, sales tax, and grad total due Double grandTotal = foodTotal's added together Double orderTax = tax * grandTotal; grandTotal = grandTotal + orderTax

Class: Order

Member variable: type

Int tableNum public int sandwich int runThrou
String userInput public int garlicBread String toppin
Public int pizza public int fountainDrink String popType
Public int salad public int toppingNumber String meat

Method name (input type): return type

Public void order(int table [][]) {
//prompt for table number
//loop for food item order placement
Do { } while (!userInput.equals("f"));
//option menu

//methods prompt for # of item ordered and add to order Public void food (int table [][], int tableNum) { Food = scanner.nextInt(); Table [tableNum] [#] = food; }

Class:

Replenish extends Kitchen

Member variable: type

String userChoice; Int userNumber;

Method name (input type): return type

//prompts user for item to restock & adjusts quantities Public void replenishStock() { //stacked if blocks based upon each food item If (userChoice.equals("food")) { userNumber = scanner.nextInt(); Food += userNumber; }

//all under do while loop Do { } while (!userChoice.equals("No"); }

to culate final tue based ered.