**Interfaces**

Create an interface Deliverable in com.pune.pizzahut package having following data

deliveryAreaLimit int (Eg. 2 km where home delivery is available)

Define a method boolean delivery()

Create a class Order implementing Deliverable with following private variables and methods

orderNo int

orderDate Date

cost int

custName String

custAddress String

approxDistance int

* Define parameterized constructor to initialize all variables.
* Override delivery() to define the delivery mode. Method returns true if approx. distance matches with the deliveryAreaLimit. Else display a message “Home Delivery not available for this distance.”

Create a class Reception having main method and perform following functionality in it

1. Accept values from user to create an order object with appropriate values
2. Reply to the user request by calling delivery() method. i.e. whether order is deliverable or not.

**[ Day 4 ]**

**Exception Handling**

Create a class InvalidPizzaTypeException and InvalidPizzaSizeException which will be user defined exception.

Modify Pizza class created in previous day assignment as bellow

* In case validation for type fails in constructor, throw a user defined checked exception called

InvalidPizzaTypeException.

* In case validation for size of pizza fails, throw a user defined checked exception called

InvalidPizzaSizeException.

* Handle these exceptions in main.

Create a class Bill with following private variables and methods

billNo int

custName String

date Date

total int

items String[ ]

Create getters and setters for all of the variables.

Create a class BillReports with following private variables and public methods

Bill bills[] i.e. array of type bill

Define methods as follows

void addBill(Bill) method to add bill object into an array

boolean updateBill(billNo,newTotal) accepts billNo and modifies the total of particular bill,

returns true if successfully updated

Bill[ ] getBills() return an array of type Bill

Bill searchBill(billNo) accepts bill no and returns details of Bill.

* In searching, in case Bill is not found in array throw user defined exception BillNotFoundException. Handle this exception in main.

Create a class Reports having main method with following functionality

1. Accept values from user and create object of Bill.
2. Store objects into array of BillReports class by using addBill() method.
3. Print details of an array by using getBills() method.
4. Accept billNo from the user along with modified bill amount. Find corresponding bill in array and update the amount. Print updated details. Use updateBill() for achieving this.
5. Accept billNo from the user and print details of bill. Use serachBill() method for achieving this.