

```

import java.text.NumberFormat;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Random;
import java.util.Set;

import javax.swing.JOptionPane;

public class JavaZon
{

    private ArrayList<Clerk> clerksList; // Arraylist to hold all the clerks
    private HashMap<String, Order> jvzOrder; // Hashmap to hold orders the key
    ...
    ...

    public JavaZon()
    {
        // initialize the Clerk
        clerksList = new ArrayList<Clerk>();
        // initialize the orders
        jvzOrder = new HashMap<String, Order>();

        // load clerks
        loadClerkList();
    }

    public void addOrder(Order anOrder)
    {
        jvzOrder.put(anOrder.getOrderID(), anOrder);
        // TODO: Code to add an order to the HashMap
        // USE THE jvzOrder object and add a new order to the hashmap
        // you need to use the orderid and the order object (anOrder)
        // The order id value is located in the order object

    }

    public void processOrder(String anID)
    {
        Order selectedOrder = jvzOrder.get(anID);

        selectedOrder.calcSubtotal();
        selectedOrder.calcTax();
        selectedOrder.calcTotal();
    }

    public void loadClerkList()
    {
        Clerk musicClerk = new Clerk("EMP100", 1, "Homer", "Simpson", "243 North Main"...
            "909-987-6666");
        Clerk bookClerk = new Clerk("EMP200", 2, "Bart", "Simpson", "454 North Euclid"...
            "909-987-4444");
        Clerk gameClerk = new Clerk("EMP300", 3, "Lisa", "Simpson", "767 North Holt", ...
            "909-987-3333");

        clerksList.add(musicClerk);
        clerksList.add(bookClerk);
        clerksList.add(gameClerk);
    }
}

```

```

}

public void setClerk(String anID)
{
    Random rand = new Random();
    int i = rand.nextInt(3);

    Order selectedOrder = jvzOrder.get(anID);
    selectedOrder.setOrderClerk(clerksList.get(i));
}

public String getReceipt(String anID)
{
    String result;

    Order selectedOrder = jvzOrder.get(anID);

    result = "JVAZON WHOLESALE STORE\n\n";

    result += "90404 CENTRAL AVE\n";
    result += "MONTCLAIR, CA, 91763\n";
    result += "  (909)345-9876\n\n";

    result += selectedOrder.toString();

    return result;
}

public String getOrdersSummary()
{
    String result = "";
    Order selectedOrder;
    Customer selectedCustomer;
    Set<String> keySet = jvzOrder.keySet();
    Object[] yourOrders = keySet.toArray();
    for(int i = 0; i < yourOrders.length; i++)
    {
        selectedOrder = jvzOrder.get(yourOrders[i]);
        selectedCustomer = selectedOrder.getCustomer();
        result += "Order ID: " + selectedOrder.getOrderID() + ", ";
        result += "Customer Name: " + selectedCustomer.getFirstName() + " ";
        result += selectedCustomer.getLastName() + ", ";
        result += selectedOrder.getTotal() + "\n";
    }
    // TODO
    // For Loop through the orders in the HashMap
    // Retrieve each order and store it in the selectedOrder object
    // retrieve the customer using the getCustomer method of the
    // selectedOrder
    // Store the customer in the selectedCustomer object
    // retrieve the customer first/last name from the selectCustomer Object
    // retrieve the order total from the selectedOrder object
    // return a String result that has all the orders in the HashMap
    // See screen printout for an example of what the result should look
    // like

    return result;
}

public String deleteOrder(String anID)
{
    String result = "";

```

```

        // TODO
        // check if the HashMap contains the order
        // If it does then remove that order from the hashMap
        // Return a message that displays the OrderID has been
        // Deleted Or not found
        // put the result from the delete into the string result

        return result;
    }

    public Order findOrder(String anID)
    {
        Order selectedOrder = jvzOrder.get(anID);

        // TODO
        // check if the HashMap contains the order
        // If it does then store the order in selectedOrder and return
        // selectedOrder
        // If the order is not found then return a null

        return selectedOrder;
    }

    public void clearOrders()
    {
        // TODO
        // Clear the hashmap of all orders
        jvzOrder.clear();
    }
}

```