Exercises based on dr.java

1.

Create a class Item that has itemcode,itemName,price and stcok.

There is a variable called count that keeps count of items that exist.

Create constructors to initialize the ItemObjects.

Generate getters of all.

Setters for those that are to be updated.

B   Create a class CartItem that extends from Item

And has following fields: itemQuantity and amount.

Generate constructors ,getters and setters and test the classes.

2

Create a class Customer has stores Customer details.

Create class Account stores Customer reference,balance and type of account.

Generate constructors,getters and setters for both the classes.

In the interactions window create an object for Account with initializing values.

Print the customer details with the Account object.

Create a class Bank that stores array of Accounts.

5 accounts are created by default as soon as the Bank Object is created.

A write a method getAccount(int accNo) that returns the Account object with accountNo as parameter.

B write a method updateBalance(int accNo,double amount) that updates the balance in the account based on the accountNo.

C

Create a class called Admin that has a method printCustomerDetails(Customer c) reads Customer reference as parameter and prints the details of the Customer.

i What type of method would be printCustomerDetails(..)

Instance or static.. justify.

D

Create an abstract class Payment that has a method makePayment();

```
public abstract class Payment{

public abstract String makePayment(double amount);

}
```

I create an object of Payment class and observe the results.

Ii Create two implementation classes CashPayment and CreditCardPayment.

```
Class CashPayment extends Payment{  }

Class CreditCardPayment extends Payment{  }
```

Compile the class and note the result.

Iii add this method inside CashPayment class

```
public String makePayment(double amount)
```

```
{
        return “amount of”+amount+”made by cash”;
}
```

lv

add this method inside CreditCardPayment class

```
        public String makePayment(double amount)
{
        return “amount of”+amount+”made by creditcard”;
}
```

Write the following code in the interactions window one by one and test them.

```
Payment ref
ref=new CashPayment()
ref.makePayment(3000)
ref=new CreditCardPayment()
ref.makePayment(5000)
```

and note the results.