

# CSE 102L Computer Programming Laboratory – Quiz01

Start: 09:00, Due Date: 10:20 Friday April 17<sup>th</sup>

---

## Disclosure

You will submit your file to an assignment that is given through MS teams. Your filename should be *Quiz01\_yourStudentNumber.java*. You are not required to provide a main method, however you can write one to test your code. Submissions made after the deadline will not be accepted, be sure to submit your work before the due date. Your code will be automatically controlled, so be sure to have the same class, method, variable names. Failure to do so may result in you receiving 0 from this quiz. All classes should be written to a single Java file. **Remove the package line if your editor adds automatically. DO NOT WRITE CLASSES IN OTHER CLASSES. THERE CAN BE ONLY 1 PUBLIC CLASS IN A SINGLE JAVA FILE.**

## Interfaces:

1. **Item** extends **Comparable** with *Item* according to price in increasing order
  - + **getPrice(): None**  
Method to return the price of *item*
  - + **getName(): String**  
Method to return the name of the *item*
2. **PaymentMethod**
  - + **pay(cart: List<Item>): boolean**  
A way of payment. Gets and sums the prices of each item in the cart, clear the cart, display the total and payment method's name. "Paid: " + total + " with " + {class name of the payment method} + *username/holderName*
3. **Colorable**
  - + **paint(color: String): None**  
Sets the color of the object.

# CSE 102L Computer Programming Laboratory – Quiz01

Start: 09:00, Due Date: 10:20 Friday April 17<sup>th</sup>

---

## Classes:

1. **Product** an abstract class that implements **Item** interface

- **name:** *String*  
name of the product
- **price:** *double*  
price of the product

+ **Product**(name: *String*, price *double*)

2-arg constructor that sets the data fields and may throw **InvalidPriceException**

+ Mutators for data fields. **setPrice** may throw **InvalidPriceException** if price is less than zero.

2. **Tax** a concrete class that implements **Item** interface

- **taxRate:** *double*  
tax rate between 0 - 1
- **item:** *Item*  
An item

+ **Tax**(taxRate: *int*, item: *Item*)

2-arg constructor that sets the data fields. **taxRate** is tax rate between 0 - 100

+ **getPrice():** *double*  
price + price \* **taxRate**

3. **Discount** a concrete class that implements **Item** interface

- **percent:** *double*  
discount percent between 0 - 1
- **item:** *Item*  
An item

+ **Discount**(percent: *int*, item: *Item*)

2-arg constructor that sets the data fields. **percent** is discount percent between 0 - 100

+ **getPrice():** *double*  
price - price \* **percent**

# CSE 102L Computer Programming Laboratory – Quiz01

Start: 09:00, Due Date: 10:20 Friday April 17<sup>th</sup>

---

## 4. **CreditCard** a concrete class that implements **PaymentMethod** interface

- **cardNumber: Long**  
16 digit card number
- **holderName: String**  
name and surname of the owner of the card
- **expirationDate: Date (java.util, not sql)**  
date of expiration of card
- **cvv: int**  
3 digit code

+ **CreditCard(parameters are above attributes in the same order)**

4-arg constructor that may throw **CreditCardException**

- **setCardNumber(cardNumber: Long): None**  
Mutator that may throw **CreditCardException** if **cardNumber** is not 16 digit

- **setCvv(cvv: int): None**

Mutator that may throw **CreditCardException** if **cvv** is not 3 digit

+ **pay(cart: List<Item>): boolean**

Taxes each item by 4%, implements according to interface description

## 5. **PayPal** a concrete class that implements **PaymentMethod** interface

- **username: String**  
username
- **password: String**  
password

+ **PayPal(username: String, password: String)**

2-arg constructor that sets data fields

+ **pay(cart: List<Item>): boolean**

Discounts each item by 6%, implements according to interface description

# CSE 102L Computer Programming Laboratory – Quiz01

Start: 09:00, Due Date: 10:20 Friday April 17<sup>th</sup>

---

## 6. Customer a concrete class

- **name:** *String*  
name of the customer
- **paymentMethod:** *PaymentMethod*  
preferred payment method of customer
- **cart:** *List<Item>*  
Collection of items in the cart of customer

### + **Customer**(name: *String*)

1-arg constructor that takes the name of the customer and initializes the empty cart as **ArrayList**

+ Accessor for **name**

+ Mutator for **paymentMethod**

+ **addItem**(item: *Item*): *boolean*

If item is **Colorable**, discounts by 8%. If item's price is > 500 applies additional tax of 20%. Applies 12% tax to each item. Adds the given item to cart

+ **removeItem**(item: *Item*): *boolean*

Removes the given item from the cart

+ **purchase**(): *boolean*

Pays with the preferred payment method

+ **display**(): *None*

Sorts and displays the cart in increasing order. {price - name}

# CSE 102L Computer Programming Laboratory – Quiz01

Start: 09:00, Due Date: 10:20 Friday April 17<sup>th</sup>

---

## Exceptions:

1. **CreditCardException** a concrete checked exception class
  - **cardNumber:** *Long*
  - **cvv:** *int*
  - + **CreditCardException(cvv: int)**
    - 1-arg constructor that sends the message “ERROR: Invalid cvv ” + **cvv**
  - + **CreditCardException(cardNumber: Long)**
    - 1-arg constructor that sends the message “ERROR: Invalid card number ” + **cardNumber**
  - + Accessors for data fields
2. **InvalidPriceException** a concrete unchecked exception class
  - **price:** *double*
  - + **InvalidPriceException(price: double)**
    - 1-arg constructor that sends “ERROR: Invalid price: ” + **price**
  - + Accessor for data field