**Arab Academy for Science and Technology and Maritime Transport**

**College of Computing and Information Technology**

| | |
|---|---|
| Course | Object-Oriented Programming (CS243) |
| Lecturer | Dr. Shaimaa ElMorsy |
| TA(s) | Mahmoud ElMorshedy, Nada Adel, Shaimaa Ahmed, Sara, Mohamed yasser |

# 12th Project

In this project, you should implement a simple ordering system that:

- displays a menu of items to the user.
- takes input from the user that represents each item and its quantity.
- adds the items with their quantities to the user's shopping cart.
- on checkout, takes input from the user representing their payment method (PayPal or CreditCard).
- Calculates and displays the amount of the order to the user and performs the payment using the method determined by the user and displays a confirmation message of the payment status (success or failure).

Design the UML class diagram of the following classes and use the Java API to implement them. (Ensure encapsulation is implemented in all classes)

- interface PaymentMethod which has the following:
  - public boolean isValid();
  - public String pay(int amount);

- Class PayPal that implements the PaymentMethod interface and has the following:
  - String variable representing the PayPal account's email.
  - String variable representing the PayPal account's password.
  - int variable representing the account's balance.
  - Constructor that takes email and password and initializes the member variables with the given parameters.
  - Mutators and accessors for all member variables.
  - isValid() method implementation (of the PaymentMethod interface) that validates the account's email and password against the following criteria:
    - email should be valid of the form (abc@def.xyz)
    - password should at least 8 characters long and has at least 1 digit, 1 lower case, 1 upper case and 1 special character.
  - pay(int amount) method implementation (of the PaymentMethod interface) that deducts the given amount from the payment method's balance if the payment method is valid and returns a success message, otherwise, returns a failure message.

- Class CreditCard that implements the PaymentMethod interface and has the following:
  - String variable representing the card's holder name.
  - String variable representing the cardNumber.
  - int variable representing the cvv.
  - Date (java.util.Date) variable representing the expiryDate.
  - int variable representing the balance.
  - Constructor that takes name, cardNumber, cvv, and expiryDate and initializes the member variables with the given parameters.
  - Mutators and accessors for all member variables.
  - isValid() method implementation (of the PaymentMethod interface) that validates the card's expiryDate is after current date.
  - pay(int amount) method implementation (of the PaymentMethod interface) that deducts the given amount from the payment method's balance if the payment method is valid and returns a success message, otherwise, returns a failure message.

- Class Item that has the following:
  - String variable representing the item's name.
  - int variable representing the item's quantity.
  - double variable representing the item's price.
  - Constructor that takes name, code, and price and initializes the member variables with the given parameters.
  - Mutators and accessors for all member variables.

- Class ShoppingCart that has the following:
  - An Array of Item objects representing the orderItems.
  - A default ShoppingCart() constructor that constructs a new empty items list.
  - public void add(Item item) that adds the given item to the list of order items, if the list is not full (i.e., has fewer than 10 items).
  - public void remove(Item item) that removes the given item from the list of order items, if the list is not empty.
  - public double getTotalAmount() that returns the total cost of all items in the orderItems list.
  - public String checkout(PaymentMethod paymentMethod) that:
    - gets the total amount of the order items.
    - calls the pay(amount) method of the paymentMethod object and returns its return value.

- Write a test program where you should do the following:
  - Create a new ShoppingCart object.
  - Display a menu of items to the user using print statements. Use the following menu as an example:

```
Please, select a product:

-  Pizza, $40
-  Cheeseburger, $20
-  Coffee, $5
-  Soda, $4
-  Water, $2
```

  - Using JOptionPane, take input from the user that represents item's name, price, and quantity.
  - Instantiate a new Item object with the given input and add the item to the shopping cart object.
  - Prompt the user with a choice to checkout by displaying a message "Proceed to checkout (Y/N)", and if the user enters "N", repeat the last two steps, otherwise, proceed to checkout.
  - On checkout, prompt the user to choose a payment method (PayPal or CreditCard) and then take the payment method information from the user accordingly (email, password, and balance) in the case of PayPal and (name, cardNumber, cvv, expiryDate, and balance) in the case of credit card.
  - Create a PaymentMethod object according to the user's choice and using the payment method information the user provides.
  - Checkout by invoking the shopping cart object's checkout(paymentMethod) method and display to the user the checkout return value.

## Rubric (10 Marks):

(2 Marks) UML class diagram.

(5 Marks) Implementation of the classes.

(3 Marks) Test program implementation and overall accuracy.

## 3 Marks Bonus for:

Any addition of the following abstract classes and/or polymorphism

- Group: 1-3 Max
- Due: Week 14
- Deliverables: A document file reporting your project that includes:
  a) Source code
  b) UML diagram of the project depicting the implemented classes and their OO relationships.
  c) Screenshots of the run of your program.

- Submit the source files (.java) and the report file to your classroom.