Universiti Malaysia PAHANG	MATAPELAJARAN: OBJECT ORIENTED PROGRAMMING			MARKAH:
	TOPIK: CLASS RELATIONSHIPS, INHERITANCE,POLYMORPHISM		KOD:	
	PENILAIAN:	BIL:	MASA:	

MATRIC NO :	SECTION:

Question:

Figure 1 shows the UML class diagram for **Kedai Saya Order System**. Kedai Saya sells football jersey to government organizations (example: schools and universities) and non government organizations (example: Football club). Based on the diagram, you are required to create Customer, Order, GovtOrder and NonGovtOrder classes.

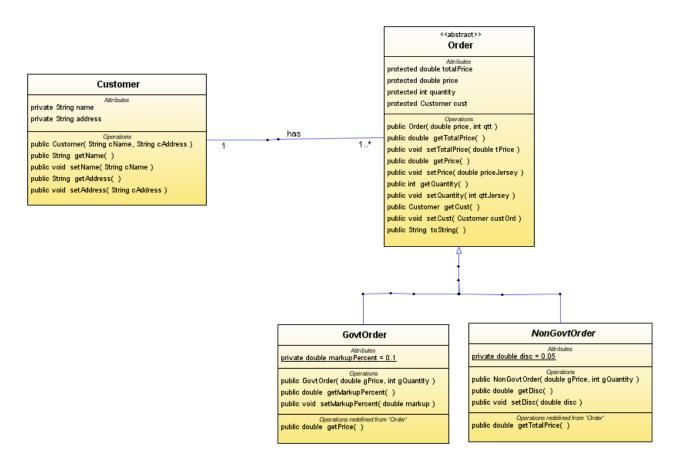


Figure 1

Universiti Malaysia PAHANG	MATAPELAJARAN: OBJECT ORIENTED PROGRAMMING			MARKAH:
	TOPIK: CLASS RELATIONSHIPS, INHERITANCE,POLYMORPHISM		KOD:	
	PENILAIAN:	BIL:	MASA:	

MATRIC NO :	SECTION:

Please complete the following steps in order to create classes in Figure 1.

- i. Create the **Customer class** in the file **Customer.java**. This class must:
 - i. Define two protected attributes; name and address
 - Define a public constructor of **Customer** that will take two arguments (customer name and customer address) so that it can be set to the name and address attributes.
 - iii. Define two public accessors; getName and getAddress to return an appropriate attribute
- ii. Create the abstract **Order class** in the file **Order.java**. This class must:
 - i. Declare two double attributes to store price and total of an order, one integer attribute to store order quantity of jersey and one customer attribute to store information about the customer who made the order.
 - Define a public constructor **Order** that will set the values of price and order quantity of jersey.
 - iii. Define four public accessors; getPrice(),getQuantity(),getCust() and getTotalPrice() to return an appropriate attribute.
 - iv. Define two public mutators; setTotalPrice() and setCust().setCust() method accepts a Customer object as its parameter and sets the data member cust to this Customer object.
 - v. Define a method toString() that returns a string representation of an object of Order class. This method display information about customer order. This information includes customer name and address, price of jersey and its quantity and also the total price of ordered jerseys.
- iii. Create the GovtOrder class that extends the Order class. This class must:

Universiti Malaysia PAHANG	MATAPELAJARAN: OBJECT ORIENTED PROGRAMMING			MARKAH:
	TOPIK: CLASS RELATIONSHIPS, INHERITANCE,POLYMORPHISM		KOD:	
	PENILAIAN:	BIL:	MASA:	

MATRIC NO :	SECTION:	

- Declare a double attribute to store rate of markup. A markup rate is a
 percentage of price increase for any purchase made by the government
 organizations (example: schools).
- ii. Define a public constructor **GovtOrder()** that specifies the order's quantity and jersey's price. This constructor calls the superclass's constructor by supplying the order's quantity and jersey's price as its arguments.
- iii. Override the method **getPrice().**This method will calculate the price of jersey after adding the markup rate. The formula for calculating price after markup is as follows:

markup = markup rate x price
price after markup = price + markup

- iv. Create the **NonGovtOrder** class that extends the **Order** class. This class must
 - i. Declare a double attribute to store percentage of discount. A discount percentage is a percentage of total price reduction for any purchase made by the non government organizations. However this disc is given for an order's quantity of more than 10 jerseys.
 - ii. Define a public constructor **NonGovtOrder()** that specifies the order's quantity and jersey's price. This constructor calls the superclass's constructor by supplying the order's quantity and jersey's price as its arguments.
 - iii. Override the method **getTotalPrice().**This method will calculate the total price of customer order and return the total price value. The total price for non government type order must include the disc percentage for an order of more than 10 jerseys only.
- v. Create a test application named **OrderSystem.java** that uses **Customer**, **Order**, **GovtOrder** and **NonGovtOrder** classes.

Universiti Malaysia PAHANG	MATAPELAJARAN: OBJECT ORIENTED PROGRAMMING			MARKAH:
	TOPIK: CLASS RELATIONSHIPS, INHERITANCE,POLYMORPHISM		KOD:	
	PENILAIAN:	BIL:	MASA:	

MATRIC NO :	SECTION:	

- Define an array of type Customer which has three elements. Name the array as cust.
- ii. Define an array of type Order which has three elements. Name the array as custOrder. Make the first array element to refer to GovtOrder class.While the second and third array elements to refer to NonGovtOrder class.
- iii. Pass the first element of **Customer** array to first **Order** array element by using **setCust()** method. Do the same for other array elements
- iv. Display the order information using **toString()** method and identify the type of order using appropriate method.
- v. Produce the sample output for this test application as in Figure 2:

```
Customer Name: Ahmad Albab
Customer Address: Universiti Malaysia Pahang, Kuantan, Pahang
Item(Jersey)
Price: 11.00
Quantity: 30
Total Price: 330.00
Type of Order: GovtOrder

Customer Name: Siti Salmah
Customer Address: Taman Tas, Kuantan, Pahang
Item(Jersey)
Price: 11.11
Quantity: 10
Total Price: 111.10
Type of Order: NonGovtOrder

Customer Name: Abu Bakar
Customer Address: Jaya Gading, Kuantan, Pahang
Item(Jersey)
Price: 11.11
Quantity: 20
Total Price: 211.09
Type of Order: NonGovtOrder

Press any key to continue . . . _
```

Figure 2

- End of Question Paper-