Data Structure and Algorithm Practicum
Class and Object

Name
Dicha Zelianivan Arkana

NIM
2241720002

Class
1i

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Question

1. Mention 2 characteristics of class/object!

   - They have data (attributes)
   - They have behaviours (methods)
   -

2. What is the keyword used to declare a class?

   In Java, we can use the `class` keyword to define a class

3. In the class **Barang** at the **Part 2**, how many attributes owned by that class? What are they?

   There are 4 attributes in class **Barang** from **Part 2** which are:

   - `String namaBarang`
   - `String jenisBarang`
   - `int stok`
   - `int hargaSatuan`

4. In the class **Barang** at the **Part 2**, on which line of code are the attributes declared?

   - `String namaBarang` - Line 4
   - `String jenisBarang` - Line 4
   - `int stok` - Line 5
   - `int hargaSatuan` - Line 5

5. In the class **Barang** at the **Part 2**, how many methods owned by that class? What are they?

   - `void tampilBarang()`
   - `void tambahStok(int n)`
   - `void tampilBarang(int n)`
   - `int hitungHargaTotal(int jumlah)`

6. In the class **Barang** at the **Part 2**, on which line of code are the methods declared?

   - `void tampilBarang()` - Line 7
   - `void tambahStok(int n)` - Line 13
   - `void tampilBarang(int n)` - Line 16
   - `int hitungHargaTotal(int jumlah)` - Line 19

7. In the method `kurangiStok()` in class **Barang**, modify the method so that it will check the availability of **stok** before subtracting the **stok**! Then it will not be any subtraction if the **stok** is already less then or equals to zero.

```
void kurangiStok(int n) {
    if (stok < n) return;
    stok = stok - n;
}
```

8. Please give your explanation, why is the method **tambahStok()** created with an `int` parameter? What is the use of that parameter in that method?

   The parameter `(int n)` will be used to add the stock of the goods. The int data type is used because we store the stock in `int`, if we use other datatype such as `double` or `float` then we need to also adjust the parameter, or type cast it.

9. Why does method **hitungHargaTotal()** have a non-void (int) data type? What is it for?

   We use a non-void, in this case an `int`, because we want to receive the total price of the goods from the caller, in this case the caller is the main method.

10. Why does the method **tambahStok()** have void data type?

    Because we only need to add the goods stock, we don't need to receive the stock amount from the caller.

11. In class **BarangMain**, in **Part 3**, on which line of code does the instantiation process run? and what is the name of the resulting object?

    The class **Barang** is instantiated in the 5th line and the object is called **b1**.

12. How do you access the attributes and methods of the objects?

    We can use the dot notation on the instance of the class. For example, if we have a property called `stock`, a method called `void addStock(int n)`,s and an object called `good`, we can do it like this:

    ```
    Good good = new Good();
    good.addStock(10); // accessing the method
    System.out.println(good.stock); // accessing the attribute
    ```

13. In class **Barang** in **Part 4**, on which line of code is the parametric constructor declared?

    The parametric constructor is declared in Line 9 through Line 14

14. In class **Barang** in **Part 4**, what does actually we do on line of code 16?

    On the 16th line, we define a method to show the goods detail.

15. Try to create another object called **b3** from class **Barang** by using the parametric constructor of class **Barang**

    ```
    Barang b3 = new Barang("QK65", "Mechanical Keyboard", 5_800_000, 10);
    ```

2 Task

1. Create the program based on the class diagram below!

Lingkaran
PHI : double r : double
hitungLuas() : double hitungKeliling() : double

Note:

- Method `hitungLuas()` will calculate the area of the the circle
- Method `hitungKeliling()` will calculate the surrounding of the circle

```
public class Lingkaran {
    public double PHI;
    public double r;

    public double hitungLuas() {
        return PHI * r * r;
    }

    public double hitungKeliling() {
        return PHI * r * 2;
    }
}
```

2. In the video game rental and shop, the most important data that they manage is **RentalTransaction**. It contains `memberId`, `memberName`, `gameName`, `dailyPrice`, and `dayRent` (how many days it will be rent). It has a method to print the rental data and the price that should be paid by member. Please create a class diagram of the class and make the code.

RentalTransaction
memberId : String memberName : String gameName : String dailyPrice : double dayRent : int
printRentalData() : void

```
class RentalTransaction {
    public String memberId;
    public String memberName;
    public String gameName;
    public double dailyPrice;
    public int dayRent;

    public RentalTransaction(
        String memberId,
        String memberName,
        String gameName,
        double dailyPrice,
        int dayRent
    ) {
        this.memberId = memberId;
        this.memberName = memberName;
        this.gameName = gameName;
        this.dailyPrice = dailyPrice;
        this.dayRent = dayRent;
    }

    void printRentalData() {
        System.out.printf("Member ID: %s\n", memberId);
        System.out.printf("Member Name: %s\n", memberName);
        System.out.printf("Game Name: %s\n", gameName);
        System.out.printf("Daily Price: %.2f\n", dailyPrice);
        System.out.printf("Day Rent: %.2f\n", dayRent);
        System.out.print("--------------\n");
        System.out.printf("Total Price: %.2f\n", dailyPrice * dayRent);
    }
}
```

3. Implement the code of this class diagram!

Item
name : String unitPrice : int qty : int
calculateTotalPrice() : int calculateDiscount() : int calculateFinalPrice() : int

- Method `calculateTotalPrice()` will multiply the quantity of item and the unit price

- Method `calculateDiscount()` will calculate the discount, with the role:
  - If total *price* > 100.000, the discount will be 10%
  - If the total price between $50.000 - 100.000$, the discount price will be 5%
  - Method `calculateFinalPrice()` will calculate the price should be paid (total price minus discount)

```
public class Item {
    public String name;
    public int unitPrice;
    public int qty;

    public Item(String name, int unitPrice, int qty) {
        this.name = name;
        this.unitPrice = unitPrice;
        this.qty = qty;
    }

    public int calculateTotalPrice() {
        return unitPrice * qty;
    }

    public int calculateDiscount() {
        int totalPrice = calculateTotalPrice();
        if (totalPrice > 100_000) return totalPrice * 0.1;
        if (totalPrice > 50_000) return totalPrice * 0.05;
        return 0;
    }

    public int calculateFinalPrice() {
        int totalPrice = calculateTotalPrice();
        int discount = calculateDiscount();
        return totalPrice - discount;
    }
}
```

4. Implement the code of class diagram below

PacMan
y : int x : int width : int height : int
moveLeft() : int moveRight() : int moveUp() : int moveDown() : int printPosition() : void

- Attribute `x` depicts the horizontal position/coordinate of Pacman, while attribute `y` depicts the vertical coordinate
- Attribute `width` is for canvas width, and attribute `height` is for the height of the canvas
- Method `moveLeft()` will move Pacman to the left (coordinate x will decrease), while `moveRight()` will move Pacman to the right (coordinate x will increase). The value of x will range from 0 to width value
- Method `moveUp()` will move Pacman to the upper position (coordinate y will decrease), while `moveDown()` will move Pacman to the lower position (coordinate y will increase). The value of y must be between 0 to height value

```
public class Pacman {
    public int x;
    public int y;
    public int width;
    public int height;

    public int moveLeft() {
        int movedValue = x - 1;
        if (movedValue >= 0) {
            x = movedValue;
        }
        return x;
    }

    public int moveRight() {
        int movedValue = x + 1;
        if (movedValue <= width) {
            x = movedValue;
        }
        return x;
    }

    public int moveUp() {
        int movedValue = y - 1;
        if (movedValue >= 0) {
            y = movedValue;
        }
        return y;
    }

    public int moveDown() {
        int movedValue = y + 1;
        if (movedValue <= height) {
            y = movedValue;
        }
        return y;
    }

    public void printPosition() {
        System.out.println("Position X: " + x);
        System.out.println("Position Y: " + y);
    }
}
```