

Week 6

| | |
|------------|-----------------------------------|
| ▼ Subject | Object Oriented Programming |
| ▼ Lecturer | Vipkas Al Hadid Firdaus S.T. M.T. |
| ▼ Type | Assignment |
| ▼ Semester | Semester 3 |
| 📅 Time | @October 8, 2023 |

Quiz 1

1. answers

- class is a template definition of the methods and variables in a particular kind of object
- object is an instance of a class, meaning that object can have state and behaviours
- by writing this code `Laptop laptop = new Laptop();` inside the `Barang` class

2. answers

- encapsulation is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit, this is important if we need a data that user can't change by user itself
- in the information system inventory, the class that is needed to be encapsulate is `stock` since we don't want the user to change the `stock` by themselves

3. answers

- it is a relation between class that is needed to run a certain code
- between `item` and `category` is that `item` can have `category`, but `category` cannot have `item`

4. answers

a. code

```
public class Barang {
    private String name;
    private int price;
    private int stock;

    public Barang(String name, int price, int stock){
        this.name = name;
        this.price = price;
        this.stock = stock;
    }

    public void setName(String name){
        this.name = name;
    }
    public String getName(){
        return this.name;
    }

    public void setStock(int stock){
        this.stock = stock;
    }

    public int getStock(){
        return this.stock;
    }

    public void setPrice(int price){
        this.price = price;
    }

    public int getPrice(){
        return this.price;
    }

    public String info(){
        String info = "";
        info += "Name\t: " + this.name + "\n";
        info += "Price\t: " + this.price + "\n";
        info += "Stock\t: " + this.stock + "\n";
        return info;
    }
}
```

- b. by giving `private` to the `name` , `price` , `stock` and using getter and setter so that it won't be changed by the user

c. `barang` can have `category`, but `category` can't have `barang`

