Object Oriented Programming
Encapsulation

Name
Dicha Zelianivan Arkana

NIM
2241720002

Class
2i

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Practicum - Encapsulation

1.1 Questions

1. Inside the class `TestMobil`, when we add the speed for the first time, why did we get a warning "The speed can't be added because the car is off"?

   Because it's illogical to add speed when the car is off. We've added a validation rule to prevent this from happening.

2. Why does the attribute `speed` and `contactOn` have the `private` access modifier?

   Because we want to encapsulate the data so that it can only be accessed from within the class. We don't want the data to be accessed from outside the class because we can't guarantee that the data will be valid if it's directly modified.

3. Change the `Motor` class so that its maximum speed is 100!

```
public class Motor {
    private int maxSpeed = 100;
    private int speed = 0;
    private boolean contactOn = false;

    public void turnOnMachine() {
        contactOn = true;
    }

    public void turnOffMachine() {
        contactOn = false;
        speed = 0;
    }

    public void incrementSpeed() {
        if (contactOn == true) {
            speed += 5;
            if (speed > maxSpeed) {
                speed = maxSpeed;
            }
        } else {
            System.out.println("The machine is off, can't add speed!");
        }
    }

    public void decrementSpeed() {
        if (contactOn == true) {
```

```
                speed -= 5;
                if (speed < 0) {
                    speed = 0;
                }
            } else {
                System.out.println("The machine is off, can't reduce speed!");
            }
        }

        public void printStatus() {
            if (contactOn == true) {
                System.out.println("The machine is on");
            } else {
                System.out.println("The machine is off");
            }
            System.out.println("Kecepetan " + speed "\n");
        }
    }
```

1.2 Questions

1. What is `getter` and `setter`?

   `Getter` is a method that is used to get the value of an attribute, while `setter` is a method that is used to set the value of an attribute. This is useful for encapsulation because we can control how the data is accessed and modified.

2. What is the use case for the method `getSavings()`?

   To get the value of the attribute `savings`. This is because the field `savings` was set to private

3. Which method is used to add balance?

   The `deposit(float money)` is used to add balance.

4. What does the `constructor` do?

   The `constructor` is used to initialize the object when it's created. It's the method that gets called when we create a new object. Its syntax is different from other methods because it doesn't have a return type.

5. What are the rules when creating a `constructor`?

   - It must have the same name as the class.
   - It doesn't have a return type.
   - It can't use `abstract`, `static`, `final`, `syncronized` modifier

6. Can a `constructor` have a `private` modifier?

   Yes, it can. We can have multiple constructors with different access modifiers. This is so that we can have multiple ways to initialize the object and encapsulate how the object is created.

7. When do we use parameters with passing parameters?

   When we want to initialize the object with a value that's passed from outside the class.

8. What's the difference between class attribute and instantiation attribute?

   Class attribute is an attribute that's shared by all objects of the class. Instantiation attribute is an attribute that's unique to each object.

2 Task

1. Write the program below and show its output

```
1  package jobsheet3;
2
3  public class EncapTest {
4      public static void main(String[] args) {
5          EncapDemo encap = new EncapDemo();
6          encap.setName("James");
7          encap.setAge(35);
8
9          System.out.println("Name: " + encap.getName());
10         System.out.println("Age: " + encap.getAge());
11     }
12 }
```

```
Run   EncapTest
/home/elianiva/.jdks/openjdk-19.0.2/bin/java -javaagent:/home/elianiva/.local/share/JetI
Name: James
Age: 30
```

2. On the program above, we've set the age to be 35 but we got 30. Explain why!

   Because we've capped the max age to be 30 on the setter. Any value above 30 will be set to 30 by the setter `setAge(int age)`.

3. Change the program above so that it has a minimum age of 18 and a maximum age of 30!

```
public void setAge(int newAge) {
    if (newAge > 30) {
        age = 30;
    } else if (newAge < 18) {
        age = 18;
    } else {
        age = newAge;
    }
}
```

4.
```
public class Anggota {
    public String nomorKTP;
    public String nama;
    public int limitPinjaman;
    public int jumlahPinjaman;
```

```
    public Anggota(String nomorKTP, String nama, int limitPinjaman) {
        this.nomorKTP = nomorKTP;
        this.nama = nama;
        this.limitPinjaman = limitPinjaman;
    }

    public String getNama() {
        return nama;
    }

    public int getLimitPinjaman() {
        return limitPinjaman;
    }

    public int getJumlahPinjaman() {
        return jumlahPinjaman;
    }

    public void pinjam(int pinjam) {
        if (pinjam > limitPinjaman) {
            System.out.println("Maaf, jumlah pinjaman melebihi limit.");
        } else {
            jumlahPinjaman += pinjam;
        }
    }

    public void angsur(int pinjam) {
        jumlahPinjaman -= pinjam;
    }
}
```

5. Modify so that the amount of the minimum installments is 10% of the loan amount! If the amount of the installments is less than 10% of the loan amount, then show a warning message "Sorry, the installment amount is less than 10% of the loan amount".

```
public void angsur(int pinjam) {
    if (pinjam < (jumlahPinjaman * 0.1)) {
        System.out.println("Maaf, angsuran harus 10% dari jumlah pinjaman.");
    } else {
        jumlahPinjaman -= pinjam;
    }
}
```

6. Modify the class `TestKoperasi` so that it can retrieve the amount of the loan and the amount of the installments from user input!

```
public class TestKoperasi {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Anggota donny = new Anggota("111333444", "Donny", 5000000);
        System.out.println("Nama Anggota: " + donny.getNama());
        System.out.println("Limit Pinjaman: " + donny.getLimitPinjaman());

        System.out.print("Masukkan jumlah pinjaman: ");
        int pinjam = scanner.nextInt();
        donny.pinjam(pinjam);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

        System.out.print("Masukkan jumlah angsuran: ");
        int angsur = scanner.nextInt();
        donny.angsur(angsur);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());

        System.out.print("Masukkan jumlah angsuran: ");
        angsur = scanner.nextInt();
        donny.angsur(angsur);
        System.out.println("Jumlah pinjaman saat ini: " + donny.getJumlahPinjaman());
    }
}
```

```
/home/elianiva/.jdks/openjdk-19.0.2/bin/java -javaagent:/home/elianiva/.local/share/JetBrains/Toolbc
Nama Anggota: Donny
Limit Pinjaman: 5000000
Masukkan jumlah pinjaman: 4000000
Jumlah pinjaman saat ini: 4000000
Masukkan jumlah angsuran: 2000000
Jumlah pinjaman saat ini: 2000000
Masukkan jumlah angsuran: 250000
Jumlah pinjaman saat ini: 1750000
```