Object Oriented Programming
Class Relation

Name
Muhammad Baihaqi Aulia Asy'ari

NIM
2241720145

Class
2I

Department
Information Technology

Study Program
D4 Informatics Engineering

Contents

1 Experiment 1

Processor.java

```java
package classrelationship.experiment1;

public class Processor {
    private String brand;
    private double cache;

    public Processor() {
    }

    public Processor(String brand, double cache) {
        this.brand = brand;
        this.cache = cache;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public String getBrand() {
        return brand;
    }

    public void setCache(double cache) {
        this.cache = cache;
    }

    public double getCache() {
        return cache;
    }

    public void info() {
        System.out.printf("Merk Processor = %s\n", brand);
        System.out.printf("Cache Memory = %.2f\n", cache);
    }
}
```

Laptop.java

```java
package classrelationship.experiment1;

public class Laptop {
    private String brand;
    private Processor processor;

    public Laptop() {
    }

    public Laptop(String brand, Processor processor) {
        this.brand = brand;
        this.processor = processor;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public void setProcessor(Processor processor) {
        this.processor = processor;
    }

    public void info() {
        System.out.println("Merk Laptop = " + brand);
        processor.info();
    }
}
```

MainExperiment1.java

```java
package classrelationship.experiment1;

public class MainExperiment1 {
    public static void main(String[] args) {
        Processor processor = new Processor("Intel i5", 3);
        Laptop laptop = new Laptop("Thinkpad", processor);

        laptop.info();

        Processor p1 = new Processor();
        p1.setBrand("Intel i5");
        p1.setCache(4);
        Laptop l1 = new Laptop();
        l1.setBrand("Thinkpad");
        l1.setProcessor(p1);
        l1.info();
    }
}
```

Terminal

```
PS D:\Kuliah>  & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
↪  80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
↪  Kuliah_28156aa7\bin'
↪  'classrelationship.experiment1.MainExperiment1'
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 3.00
Merk Laptop = Thinkpad
Merk Processor = Intel i5
Cache Memory = 4.00
```

1.1 Question

Berdasarkan percobaan 1, jawablah pertanyaan-pertanyaan yang terkait:

1. Di dalam *class* `Processor` dan *class* `Laptop` , terdapat method *setter* dan *getter* untuk masing-masing atributnya. Apakah gunanya *method setter* dan *getter* tersebut ?

2. Di dalam *class* `Processor` dan *class* `Laptop`, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

3. Perhatikan *class* `Laptop`, di antara 2 atribut yang dimiliki (*merk* dan *proc*), atribut manakah yang bertipe *object* ?

4. Perhatikan *class* `Laptop`, pada baris manakah yang menunjukan bahwa *class* `Laptop` memiliki relasi dengan *class* `Processor` ?

5. Perhatikan pada *class* `Laptop` , Apakah guna dari sintaks `proc.info()` ?

6. Pada *class* `MainPercobaan1`, terdapat baris kode:

   ```
   Laptop l = new Laptop("Thinkpad", p);
   ```

   Apakah `p` tersebut ?
   Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

   ```
   Laptop l = new Laptop("Thinkpad", new Processor("Intel i5", 3));
   ```

   Bagaimanakah hasil program saat dijalankan, apakah ada perubahan ?

1.2 Answer

1. To set or get a value that has to follow a certain rule (encapsulation) and to set a value when the default constructor is used.

2. Default constructor makes us declare the value of the object's attribute using the setter. The parametric constructor set the value accordingly in the instantiation of the object.

3. proc is the object attribute. The attribute derived from the Processor class.

4. The line where it state the attribute of class Laptop uses the Processor class as an object attribute. In my instance is in line 5 of the class Laptop.java .

5. It is used to call the method info() in the Laptop class which give description on the instance of said Laptop object.

6. The p is variable used to store the instance of the Processor object that has been instantiated. Nothing will change in the eyes of the users if that method of instantiation is used.

2 Experiment 2

Car.java

```java
package classrelationship.experiment2;

public class Car {
    private String brand;
    private int cost;

    public Car() {
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public String getBrand() {
        return brand;
    }

    public void setCost(int cost) {
        this.cost = cost;
    }

    public int getCost() {
        return cost;
    }

    public int calculateCarCost(int day) {
        return cost * day;
    }
}
```

Driver.java

```java
package classrelationship.experiment2;

public class Driver {
    private String name;
    private int cost;

    public Driver() {
    }

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

    public String getName() {
        return name;
    }

    public void setCost(int cost) {
        this.cost = cost;
    }

    public int getCost() {
        return cost;
    }

    public int calculateDriverCost(int day) {
        return cost * day;
    }
}
```

User.java

```java
package classrelationship.experiment2;

public class User {
    private String name;
    private Car car;
    private Driver driver;
    private int day;

    public User() {
    }

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

    public String getName() {
        return name;
    }

    public void setCar(Car car) {
        this.car = car;
    }

    public Car getCar() {
        return car;
    }

    public void setDriver(Driver driver) {
        this.driver = driver;
    }

    public Driver getDriver() {
        return driver;
    }

    public void setDay(int day) {
        this.day = day;
    }

    public int getDay() {
        return day;
```

```
    }

    public int calulateTotalCost() {
        return car.calculateCarCost(day) +
        ↪  driver.calculateDriverCost(day);
    }
}
```

MainExperiment2.java

```
package classrelationship.experiment2;

public class MainExperiment2 {
    public static void main(String[] args) {
        Car car = new Car();
        car.setBrand("Avanza");
        car.setCost(350_000);

        Driver driver = new Driver();
        driver.setName("John Doe");
        driver.setCost(200_000);

        User user = new User();
        user.setName("Jane Doe");
        user.setCar(car);
        user.setDriver(driver);
        user.setDay(2);

        System.out.println("Total Cost = " +
        ↪  user.calulateTotalCost());
    }
}
```

Terminal

```
PS D:\Kuliah>  & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
↪  80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
↪  Kuliah_28156aa7\bin'
↪  'classrelationship.experiment2.MainExperiment2'
Total Cost = 1100000
```

2.1 Question

1. Perhatikan class Pelanggan. Pada baris program manakah yang menunjukan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?

2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari ?

3. Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

4. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?

5. Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?

6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main dan amati perubahan saat di-run!

   ```
   System.out.println(p.getMobil().getMerk());
   ```

   Jadi untuk apakah sintaks p.getMobil().getMerk() yang ada di dalam method main tersebut?

2.2 Answer

1. On the line where it's declaring the attribute as an object of Car and Driver. In this case, line 5 and 6.

2. Because both class don't have and don't know how many day it will be. The day attribute is owned by the User class.

3. To get the calculation on the cost of the car rent and the driver fee.

4. To set the User attribute using the instantiated car object and driver object.

5. To get the sum of all cost for the user.

6. To get the brand name of the car used by the user.

3 Experiment 3

Employee.java

```java
package classrelationship.experiment3;

public class Employee {
    private String nip;
    private String name;

    public Employee(String nip, String name) {
        this.nip = nip;
        this.name = name;
    }

    public void setNip(String nip) {
        this.nip = nip;
    }

    public String getNip() {
        return nip;
    }

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

    public String getName() {
        return name;
    }

    public String info() {
        String info = "";
        info += "NIP: " + this.nip + "\n";
        info += "Name: " + this.name + "\n";
        return info;
    }
}
```

Train.java

```java
package classrelationship.experiment3;

public class Train {
    private String name;
    private String classification;
    private Employee conductor;
    private Employee assitant;

    public Train(String name, String classification, Employee
    ↪ conductor) {
        this.name = name;
        this.classification = classification;
        this.conductor = conductor;
    }

    public Train(String name, String classification, Employee
    ↪ conductor, Employee assistant) {
        this.name = name;
        this.classification = classification;
        this.conductor = conductor;
        this.assitant = assistant;
    }

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

    public String getName() {
        return name;
    }

    public void setClassification(String classification) {
        this.classification = classification;
    }

    public String getClassification() {
        return classification;
    }

    public void setConductor(Employee conductor) {
        this.conductor = conductor;
```

```
40      }
41
42      public Employee getConductor() {
43          return conductor;
44      }
45
46      public void setAssitant(Employee assitant) {
47          this.assitant = assitant;
48      }
49
50      public Employee getAssitant() {
51          return assitant;
52      }
53
54      public String info() {
55          String info = "";
56          info += "Name      : " + this.name + "\n";
57          info += "Class     : " + this.classification + "\n";
58          info += "Conductor: " + this.conductor.info() + "\n";
59          info += "Assistant: " + this.assitant.info() + "\n";
60          return info;
61      }
62  }
```

MainExperiment3.java

```
1   package classrelationship.experiment3;
2
3   public class MainExperiment3 {
4       public static void main(String[] args) {
5           Employee conductor = new Employee("1234", "Spongebob
            ↪  Squarepants");
6           Employee assistant = new Employee("4567", "Patrick Star");
7           Train train = new Train("New Style", "Bussiness", conductor,
            ↪  assistant);
8           System.out.println(train.info());
9       }
10  }
```

Terminal

```
PS D:\Kuliah>  & 'C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe'
 ↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
 ↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
 ↪  80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
 ↪  Kuliah_28156aa7\bin'
 ↪  'classrelationship.experiment3.MainExperiment3'
Name     : New Style
Class    : Bussiness
Conductor: NIP: 1234
Name: Spongebob Squarepants

Assistant: NIP: 4567
Name: Patrick Star
```

3.1 Question

1. Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa?

2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main() !

   ```
   Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
   KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis",
    ↪  masinis);
   System.out.println(keretaApi.info());
   ```

3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

4. Perbaiki class KeretaApi sehingga program dapat berjalan !

3.2 Answer

1. To get the info of the object in a form of a String to be append in the info String of the Train info method.

2. MainQuestion.java

   ```
   package classrelationship.experiment3;

   public class MainQuestion {
       public static void main(String[] args) {
   ```

```
        Employee conductor = new Employee("1234", "Spongebob
        ↪  Squarepants");
        Train train = new Train("New Style", "Bussiness",
        ↪  conductor);
        System.out.println(train.info());
    }
}
```

3. Terminal

```
PS D:\Kuliah>  & 'C:\Program
↪  Files\Java\jdk-18.0.2.1\bin\java.exe'
↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\
↪  workspaceStorage\80d97a47d24665dc0bce7ab1e048ecbd\
↪  redhat.java\jdt_ws\Kuliah_28156aa7\bin'
↪  'classrelationship.experiment3.MainQuestion'
Exception in thread "main" java.lang.NullPointerException: Cannot
↪  invoke "classrelationship.experiment3.Employee.info()"
↪  because "this.assitant" is null
    at classrelationship.experiment3.Train.info (Train.java:59)
    at classrelationship.experiment3.MainQuestion.main
    ↪  (MainQuestion.java:7)
```

Because on the Train info() method it also ask for the data of the assistant which is nonexistance in this instance.

4. part of `Train.java`

```
public String info() {
    String info = "";
    info += "Name      : " + this.name + "\n";
    info += "Class     : " + this.classification + "\n";
    info += "Conductor: " + this.conductor.info() + "\n";
    if (assitant != null) {
        info += "Assistant: " + this.assitant.info() + "\n";
    }
    return info;
}
```

One way to make it work is to only print assistant info if assistant is not null in the info method

4 Experiment 4

Passenger.java

```java
package classrelationship.experiment4;

public class Passenger {
    private String IDCard;
    private String name;

    public Passenger(String IDCard, String name) {
        this.IDCard = IDCard;
        this.name = name;
    }

    public void setIDCard(String iDCard) {
        IDCard = iDCard;
    }

    public String getIDCard() {
        return IDCard;
    }

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

    public String getName() {
        return name;
    }

    public String info() {
        String info = "";
        info += String.format("ID Card: %s %n", IDCard);
        info += String.format("Name: %s %n", name);
        return info;
    }
}
```

Seat.java

```java
package classrelationship.experiment4;

public class Seat {
```

```
4       private String seatNumber;
5       private Passenger passenger;
6
7       public Seat(String seatNumber) {
8           this.seatNumber = seatNumber;
9       }
10
11      public void setSeatNumber(String seatNumber) {
12          this.seatNumber = seatNumber;
13      }
14
15      public String getSeatNumber() {
16          return seatNumber;
17      }
18
19      public void setPassenger(Passenger passenger) {
20          this.passenger = passenger;
21      }
22
23      public Passenger getPassenger() {
24          return passenger;
25      }
26
27      public String info() {
28          String info = "";
29          info += String.format("Seat Number: %s %n", seatNumber);
30          if (this.passenger != null) {
31              info += String.format("Passenger: %s %n", passenger);
32          }
33          return info;
34      }
35  }
```

Carriage.java

```
1   package classrelationship.experiment4;
2
3   public class Carriage {
4       private String code;
5       private Seat[] seats;
6
7       private void initSeat() {
8           for (int i = 0; i < seats.length; i++) {
```

```
            this.seats[i] = new Seat(String.valueOf(i + 1));
        }
    }

    public Carriage(String code, int amount) {
        this.code = code;
        this.seats = new Seat[amount];
        this.initSeat();
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getCode() {
        return code;
    }

    public void setPassenger(Passenger passenger, int number) {
        this.seats[number-1].setPassenger(passenger);
    }

    public Seat[] getSeats() {
        return seats;
    }

    public String info() {
        String info = "";
        info += String.format("Code: %s %n", code);
        for (Seat seat : seats) {
            info += seat.info();
        }
        return info;
    }
}
```

MainExperiment4.java

```
package classrelationship.experiment4;

public class MainExperiment4 {
    public static void main(String[] args) {
        Passenger p = new Passenger("12345", "Mr. Krab");
```

```
        Carriage carriage = new Carriage("A", 10);
        carriage.setPassenger(p, 1);
        System.out.println(carriage.info());
    }
}
```

Terminal

```
PS D:\Kuliah>  d:; cd 'd:\Kuliah'; & 'C:\Program
 ↪  Files\Java\jdk-18.0.2.1\bin\java.exe'
 ↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
 ↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
 ↪  80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
 ↪  Kuliah_28156aa7\bin'
 ↪  'classrelationship.experiment4.MainExperiment4'
Code: A
Seat Number: 1
Passenger: ID Card: 12345
Name: Mr. Krab

Seat Number: 2
Seat Number: 3
Seat Number: 4
Seat Number: 5
Seat Number: 6
Seat Number: 7
Seat Number: 8
Seat Number: 9
Seat Number: 10
```

4.1 Question

1. Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

```
...
if (this.penumpang != null) {
    info += "Penumpang: " + penumpang.info() + "\n";
}
...
```

3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1 ?

4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi ?

5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain !

4.2 Answer

1. There are 10 seats in the carriage.

2. If the passenger variable is not null add passenger info into variable info

3. Because array index start at 0 and to adjust with the size of the array with the index, when we enter the index we decrease the number by 1.

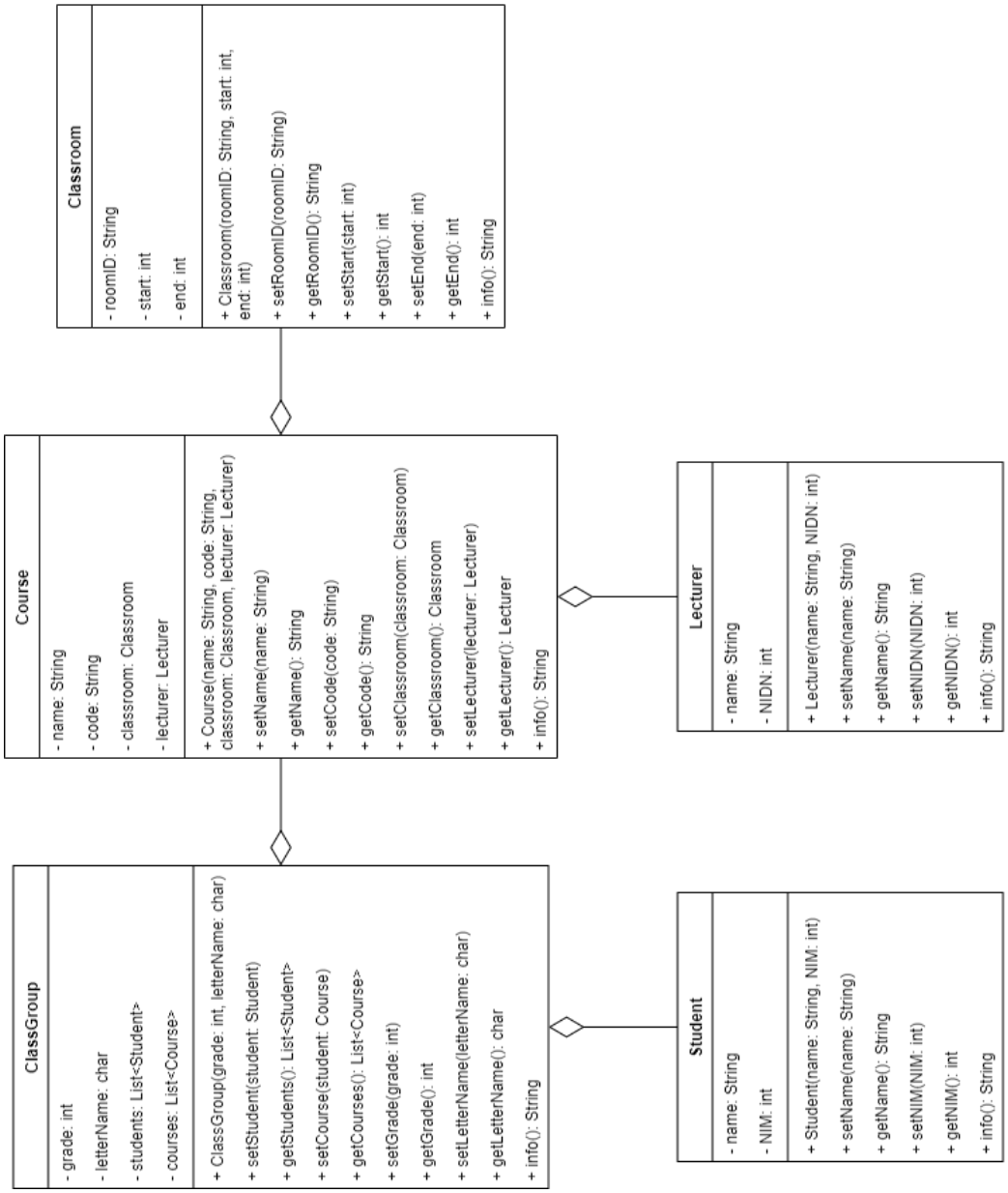
4. Mr. Krab is replaced with budi.

5. -

```
public void setPassenger(Passenger passenger, int number) {
    if (seats[number-1] == null) {
        this.seats[number-1].setPassenger(passenger);
    } else {
        System.out.println("Seat is occupied");
    }
}
```

5 Assignment

Berdasarkan latihan di pertemuan teori, rancang dengan class diagram, kemudian implementasikan ke dalam program! Studi kasus harus mewakili relasi class dari percobaan-percobaan yang telah dilakukan pada materi ini, setidaknya melibatkan minimal 4 class (class yang berisi main tidak dihitung).


Classroom
- roomID: String
- start: int
- end: int
+ Classroom(roomID: String, start: int, end: int)
+ setRoomID(roomID: String)
+ getRoomID(): String
+ setStart(start: int)
+ getStart(): int
+ setEnd(end: int)
+ getEnd(): int
+ info(): String
Course
- name: String
- code: String
- classroom: Classroom
- lecturer: Lecturer
+ Course(name: String, code: String, classroom: Classroom, lecturer: Lecturer)
+ setName(name: String)
+ getName(): String
+ setCode(code: String)
+ getCode(): String
+ setClassroom(classroom: Classroom)
+ getClassroom(): Classroom
+ setLecturer(lecturer: Lecturer)
+ getLecturer(): Lecturer
+ info(): String
Lecturer
- name: String
- NIDN: int
+ Lecturer(name: String, NIDN: int)
+ setName(name: String)
+ getName(): String
+ setNIDN(NIDN: int)
+ getNIDN(): int
+ info(): String
ClassGroup
- grade: int
- letterName: char
- students: List<Student>
- courses: List<Course>
+ ClassGroup(grade: int, letterName: char)
+ setStudent(student: Student)
+ getStudents(): List<Student>
+ setCourse(student: Course)
+ getCourses(): List<Course>
+ setGrade(grade: int)
+ getGrade(): int
+ setLetterName(letterName: char)
+ getLetterName(): char
+ info(): String
Student
- name: String
- NIM: int
+ Student(name: String, NIM: int)
+ setName(name: String)
+ getName(): String
+ setNIM(NIM: int)
+ getNIM(): int
+ info(): String

Lecturer.java

```java
package classrelationship.assignment;

public class Lecturer {
    private String name;
    private String NIDN;

    public Lecturer(String name, String NIDN) {
        this.name = name;
        this.NIDN = NIDN;
    }

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

    public String getName() {
        return name;
    }

    public void setNIDN(String NIDN) {
        this.NIDN = NIDN;
    }

    public String getNIDN() {
        return NIDN;
    }

    public String info() {
        String info = "";
        info += String.format("Name: %s %n", getName());
        info += String.format("NIDN: %s %n", getNIDN());
        return info;
    }
}
```

Student.java

```java
package classrelationship.assignment;

public class Student {
    private String name;
    private String NIM;
```

```
6
7       public Student(String name, String NIM) {
8           this.name = name;
9           this.NIM = NIM;
10      }
11
12      public void setName(String name) {
13          this.name = name;
14      }
15
16      public String getName() {
17          return name;
18      }
19
20      public void setNIM(String NIM) {
21          this.NIM = NIM;
22      }
23
24      public String getNIM() {
25          return NIM;
26      }
27
28      public String info() {
29          String info = "";
30          info += String.format("Name: %s %n", name);
31          info += String.format("NIM : %s %n", NIM);
32          return info;
33      }
34  }
```

Classroom.java

```
1   package classrelationship.assignment;
2
3   public class Classroom {
4       private String roomID;
5       private int start;
6       private int end;
7
8       public Classroom(String roomID, int start, int end) {
9           this.roomID = roomID;
10          this.start = start;
11          this.end = end;
```

```
12      }
13
14      public void setRoomID(String roomID) {
15          this.roomID = roomID;
16      }
17
18      public String getRoomID() {
19          return roomID;
20      }
21
22      public void setStart(int start) {
23          this.start = start;
24      }
25
26      public int getStart() {
27          return start;
28      }
29
30      public void setEnd(int end) {
31          this.end = end;
32      }
33
34      public int getEnd() {
35          return end;
36      }
37
38      public String info() {
39          String info = "";
40          info += String.format("Room ID: %s %n", roomID);
41          info += String.format("Session: %d - %d %n", start, end);
42          return info;
43      }
44  }
```

Course.java

```
1   package classrelationship.assignment;
2
3   public class Course {
4       private String name;
5       private String code;
6       private Classroom classroom;
7       private Lecturer lecturer;
```

```
8
9       public Course(String name, String code, Classroom classroom,
        ↪  Lecturer lecturer) {
10          this.name = name;
11          this.code = code;
12          this.classroom = classroom;
13          this.lecturer = lecturer;
14      }
15
16      public void setName(String name) {
17          this.name = name;
18      }
19
20      public String getName() {
21          return name;
22      }
23
24      public void setCode(String code) {
25          this.code = code;
26      }
27
28      public String getCode() {
29          return code;
30      }
31
32      public void setClassroom(Classroom classroom) {
33          this.classroom = classroom;
34      }
35
36      public Classroom getClassroom() {
37          return classroom;
38      }
39
40      public void setLecturer(Lecturer lecturer) {
41          this.lecturer = lecturer;
42      }
43
44      public Lecturer getLecturer() {
45          return lecturer;
46      }
47
48      public String info() {
```

```
49          String info = "";
50          info += String.format("Course Name: %s %n", name);
51          info += String.format("Course Code: %s %n", code);
52          info += String.format("Classroom  : %n%s", classroom.info());
53          info += String.format("Lecturer   : %n%s", lecturer.info());
54          return info;
55      }
56  }
```

ClassGroup.java

```
1   package classrelationship.assignment;
2
3   import java.util.ArrayList;
4   import java.util.List;
5
6   public class ClassGroup {
7       private int grade;
8       private char letterName;
9       private List<Student> students = new ArrayList<Student>();
10      private List<Course> courses = new ArrayList<Course>();
11
12      public ClassGroup(int grade, char letterName) {
13          this.grade = grade;
14          this.letterName = letterName;
15      }
16
17      public void setGrade(int grade) {
18          this.grade = grade;
19      }
20
21      public int getGrade() {
22          return grade;
23      }
24
25      public void setLetterName(char letterName) {
26          this.letterName = letterName;
27      }
28
29      public char getLetterName() {
30          return letterName;
31      }
32
```

```
33      public void setStudent(Student students) {
34          this.students.add(students);
35      }
36
37      public List<Student> getStudents() {
38          return students;
39      }
40
41      public void setCourse(Course course) {
42          this.courses.add(course);
43      }
44
45      public List<Course> getCourses() {
46          return courses;
47      }
48
49      public String info() {
50          String info = "";
51          info += String.format("Class       : %d%c %n", grade,
            ↪  letterName);
52          info += String.format("Student List: %n");
53          for (Student student : students) {
54              info += student.info();
55          }
56          info += String.format("Course List : %n");
57          for (Course course : courses) {
58              info += course.info();
59          }
60          return info;
61      }
62  }
```

MainAssignment.java

```
1   package classrelationship.assignment;
2
3   public class MainAssignment {
4       public static void main(String[] args) {
5           ClassGroup internationalClassGroup = new ClassGroup(2, 'I');
6           internationalClassGroup.setStudent(new Student("Alpha",
            ↪  "2241720001"));
7           internationalClassGroup.setStudent(new Student("Beta",
            ↪  "2241720002"));
```

```
        internationalClassGroup.setStudent(new Student("Charlie",
        ↪  "2241720003"));
        internationalClassGroup.setStudent(new Student("Delta",
        ↪  "2241720004"));
        internationalClassGroup.setStudent(new Student("Echo",
        ↪  "2241720005"));
        internationalClassGroup.setStudent(new Student("Foxtrot",
        ↪  "2241720006"));

        Lecturer lecturer = new Lecturer("Vipkas Al Hadid Firdaus,
        ↪  S.T., M.T.", "0005059104");
        Classroom LSI2 = new Classroom("LSI2_6T", 8, 11);
        Course OOP = new Course("Object Oriented Programming",
        ↪  "RTI223007", LSI2, lecturer);
        internationalClassGroup.setCourse(OOP);

        Classroom LERP = new Classroom("LERP_7T", 8, 11);
        Course PracticumOOP = new Course("Practicum Object Oriented
        ↪  Programming", "RTI223008", LERP, lecturer);
        internationalClassGroup.setCourse(PracticumOOP);

        System.out.println(internationalClassGroup.info());
    }
}
```

Terminal

```
PS D:\Kuliah>  d:; cd 'd:\Kuliah'; & 'C:\Program
↪  Files\Java\jdk-18.0.2.1\bin\java.exe'
↪  '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
↪  'C:\Users\G4CE-PC\AppData\Roaming\Code\User\workspaceStorage\
↪  80d97a47d24665dc0bce7ab1e048ecbd\redhat.java\jdt_ws\
↪  Kuliah_28156aa7\bin' 'classrelationship.assignment.MainAssignment'
Class       : 2I
Student List:
Name: Alpha
NIM : 2241720001
Name: Beta
NIM : 2241720002
Name: Charlie
NIM : 2241720003
Name: Delta
NIM : 2241720004
```

```
Name: Echo
NIM : 2241720005
Name: Foxtrot
NIM : 2241720006
Course List :
Course Name: Object Oriented Programming
Course Code: RTI223007
Classroom  :
Room ID: LSI2_6T
Session: 8 - 11
Lecturer   :
Name: Vipkas Al Hadid Firdaus, S.T., M.T.
NIDN: 0005059104
Course Name: Practicum Object Oriented Programming
Course Code: RTI223008
Classroom  :
Room ID: LERP_7T
Session: 8 - 11
Lecturer   :
Name: Vipkas Al Hadid Firdaus, S.T., M.T.
NIDN: 0005059104
```