# Object Oriented Programming
# Quiz 2


**Name**
Muhammad Baihaqi Aulia Asy'ari

**NIM**
2241720145

**Class**
2I

**Department**
Information Technology

**Study Program**
D4 Informatics Engineering

# 1 Quiz 2 Object Oriented Programming

1. Student.java

```java
import java.util.List;

public class Student {
    // defining class fields
    private String name;
    private int nim;
    private List<Double> grades;

    // defining constructor
    public Student(String name, int nim, List<Double> grades) {
        this.name = name;
        this.nim = nim;
        this.grades = grades;
    }

    // field getter setter
    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setNim(int nim) {
        this.nim = nim;
    }

    public int getNim() {
        return nim;
    }

    public void setGrade(List<Double> grades) {
        this.grades = grades;
    }

    public List<Double> getGrade() {
        return grades;
    }
```

```
40
41      // defining method to calculate GPA
42      public double calcualteGPA() {
43          double sum = 0;
44          for (double grade : getGrade()) {
45              sum += grade;
46          }
47          return sum/getGrade().size();
48      }
49
50      // overload method with new param
51      public double calcualteGPA(List<Double> credits) {
52          double sum = 0;
53          for (int i = 0; i < getGrade().size(); i++) {
54              sum += getGrade().get(i) * credits.get(i);
55          }
56          return sum/getGrade().size();
57      }
58  }
```

2. Subject.java

```
1   import java.util.List;
2
3   public class Subject {
4       // defining class fields
5       private String subjectName;
6       private double credit;
7       private List<Double> studentsGrade;
8
9       // defining constructor
10      public Subject(String subjectName, double credit,
        ↪  List<Double> studentsGrade) {
11          this.subjectName = subjectName;
12          this.credit = credit;
13          this.studentsGrade = studentsGrade;
14      }
15
16      // field getter setter
17      public void setSubjectName(String subjectName) {
18          this.subjectName = subjectName;
19      }
20
```

```
21      public String getSubjectName() {
22          return subjectName;
23      }
24
25      public void setCredit(double credit) {
26          this.credit = credit;
27      }
28
29      public double getCredit() {
30          return credit;
31      }
32
33      public void setStudentsGrade(List<Double> studentsGrade) {
34          this.studentsGrade = studentsGrade;
35      }
36
37      public List<Double> getStudentsGrade() {
38          return studentsGrade;
39      }
40
41      // defining method to calculate weight
42      public double calculateWeight() {
43          double sum = 0;
44          for (int i = 0; i < studentsGrade.size(); i++) {
45              sum += studentsGrade.get(i);
46          }
47          return sum/studentsGrade.size();
48      }
49      // overload method with new param
50      public double calculateWeight(List<Double> studentsWeight) {
51          double sum = 0;
52          for (int i = 0; i < studentsGrade.size(); i++) {
53              sum += studentsGrade.get(i) * studentsWeight.get(i);
54          }
55          return sum/studentsGrade.size();
56      }
57  }
```

3. Representative.java

```
1   import java.util.List;
2
3   public class Representative {
```

```
4       // defining class fields
5       private List<Student> students;
6       private List<Subject> subjects;
7
8       // field getter setter
9       public void setStudents(List<Student> students) {
10          this.students = students;
11      }
12
13      public List<Student> getStudents() {
14          return students;
15      }
16
17      public void setSubjects(List<Subject> subjects) {
18          this.subjects = subjects;
19      }
20
21      public List<Subject> getSubjects() {
22          return subjects;
23      }
24
25      // define method to list student and subject
26      public void displayStudentList() {
27          System.out.println("Student and Subject List");
28          for (int i = 0; i < getStudents().size(); i++) {
29              String name = getStudents().get(i).getName();
30              double nim = getStudents().get(i).getNim();
31              double grade = getStudents().get(i).calcualteGPA();
32              String subjectName =
                ↪  getSubjects().get(i).getSubjectName();
33              double credit = getSubjects().get(i).getCredit();
34              double weight =
                ↪  getSubjects().get(i).calculateWeight();
35
36              System.out.printf("Student  : %s %n", name);
37              System.out.printf("  NIM    : %.2f %n", nim);
38              System.out.printf("  GPA    : %.2f %n", grade);
39              System.out.printf("Subject  : %s %n", subjectName);
40              System.out.printf("  Credit : %.2f %n", credit);
41              System.out.printf("  Weight : %.2f %n", weight);
42          }
43      }
```

```
}
```

4. Main.java

```
import java.util.ArrayList;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        // define grades
        List<Double> grades1 = new ArrayList<Double>();
        grades1.add(90.0);
        grades1.add(87.0);
        grades1.add(80.0);
        List<Double> grades2 = new ArrayList<Double>();
        grades2.add(80.0);
        grades2.add(90.0);
        grades2.add(87.0);

        // define students
        Student student1 = new Student("student 1", 0001,
        ↪  grades1);
        Student student2 = new Student("student 2", 0002,
        ↪  grades2);

        // add students to list
        List<Student> students = new ArrayList<Student>();
        students.add(student1);
        students.add(student2);

        // define subjects
        Subject subject1 = new Subject("subject 1", 0.5,
        ↪  grades1);
        Subject subject2 = new Subject("subject 2", 0.6,
        ↪  grades2);

        // add subjects to list
        List<Subject> subjects = new ArrayList<Subject>();
        subjects.add(subject1);
        subjects.add(subject2);

        // add students and subjects list to representative
        Representative representative = new Representative();
```

```
36          representative.setStudents(students);
37          representative.setSubjects(subjects);
38
39          // display student list
40          representative.displayStudentList();
41      }
42  }
```

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' 'Main'
Student and Subject List
Student  : student 1
NIM    : 1.00
GPA    : 85.67
Subject  : subject 1
Credit : 0.50
Weight : 85.67
Student  : student 2
NIM    : 2.00
GPA    : 85.67
Subject  : subject 2
Credit : 0.60
Weight : 85.67
```