Object Oriented Programming
Inheritance

Name
Dicha Zelianivan Arkana

NIM
2241720002

Class
2i

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Questions

1. **ClassA.java**

```
public class ClassA {
    public int X;
    public int Y;

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

ClassB.java

```
public class ClassB extends ClassA {
    public int z;

    public void getNilaiZ(){
        System.out.println("nilai z : " + z);
    }

    public void getJumlah(){
        System.out.println("jumlah : " + (X + Y + z));
    }
}
```

Percobaan1.java

```
public class Percobaan1 {
    public static void main(String[] args) {
        ClassB hitung = new ClassB();
        hitung.X = 20;
        hitung.Y = 30;
        hitung.z = 5;
        hitung.getNilai();
        hitung.getNilaiZ();
        hitung.getJumlah();
    }
}
```

Output

```
nilai x : 20
nilai y : 30
nilai z : 5
jumlah : 55
```

2. It was throwing an error because the `ClassB` wasn't inheriting the `ClassA`.

2 Questions

1. **ClassA.java**

```
public class ClassA {
    protected int X;
    protected int Y;

    public void setX(int x) {
        X = x;
    }

    public void setY(int y) {
        Y = y;
    }

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

ClassB.java

```
public class ClassB extends ClassA {
    private int z;

    public void setZ(int z) {
        this.z = z;
    }

    public void getNilaiZ(){
        System.out.println("nilai z : " + z);
    }

    public void getJumlah(){
```

```
        System.out.println("jumlah : " + (X + Y + z));
    }
}
```

Percobaan2.java

```
public class Percobaan2 {
    public static void main(String[] args) {
        ClassB hitung = new ClassB();
        hitung.setX(20);
        hitung.setY(30);
        hitung.setZ(5);
        hitung.getNilai();
        hitung.getNilaiZ();
        hitung.getJumlah();
    }
}
```

Output

```
nilai x : 20
nilai y : 30
nilai z : 5
jumlah : 55
```

2. Because the access modifier was `private`, it can't be accessed by the child class. We need to change it to `protected` so that it can be accessed by the child class.

3 Questions

1. `ClassA` is the superclass of both `ClassB` and `ClassC` while `ClassB` is the superclass of `ClassC`. `ClassC` is the subclass of both `ClassB` and `ClassA` while `ClassB` is the subclass of `ClassA`.

2. The constructor will be called from the superclass to the subclass. The `super()` invocation must be done in the first line of the constructor. Which is why it throws an error when we put it in the last line of the constructor.

3. It calls the parent class or the superclass constructor.

4 Task

- **Employee.java**

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

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

    public String getName() {
        return name;
    }

    public int getSalary() {
        return 0;
    }
}
```

- **Lecturer.java**

```
public class Lecturer extends Employee {
    private int creditsCount;
    private int creditValue;

    public Lecturer(String nip, String name, String address) {
        super(nip, name, address);
    }

    public void setCreditsCount(int creditsCount) {
        this.creditsCount = creditsCount;
    }

    public void setCreditValue(int creditValue) {
        this.creditValue = creditValue;
    }

    @Override
    public int getSalary() {
        return creditsCount * creditValue;
```

```
    }
}
```

- **Payroll.java**

```
public class Payroll {
    private final List<Employee> employees = new ArrayList<>();

    public void addEmployee(Employee employee) {
        employees.add(employee);
    }

    public void printPayroll() {
        for (Employee employee : employees) {
            System.out.println(employee.getName() + " " + employee.getSalary());
        }
    }
}
```

- **TaskMain.java**

```
public class TaskMain {
    public static void main(String[] args) {
        Payroll payroll = new Payroll();
        Lecturer lecturer = new Lecturer("12345678", "Manusia Bernapas", "Mosque Stree
        lecturer.setCreditsCount(10);
        lecturer.setCreditValue(100000);
        payroll.addEmployee(lecturer);
        payroll.printPayroll();
    }
}
```

- **Output**

```
Manusia Bernapas 1000000
```