Object Oriented Programming
Overloading and Overriding

Name
Muhammad Baihaqi Aulia Asy'ari

NIM
2241720145

Class
2I

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Experiment 1

Main.java

```java
package experiment1;

public class Main {
    public static void main(String[] args) {
        Manager[] managers = new Manager[2];
        Staff[] managerOneStaffs = new Staff[2];
        Staff[] managerTwoStaffs = new Staff[3];

        // Managers Instatiation

        managers[0] = new Manager();
        managers[0].setName("Alpha");
        managers[0].setNip("230001");
        managers[0].setBracket("1");
        managers[0].setBonus(5_000_000);
        managers[0].setSection("Alpha Squad");

        managers[1] = new Manager();
        managers[1].setName("Delta");
        managers[1].setNip("230004");
        managers[1].setBracket("1");
        managers[1].setBonus(2_500_000);
        managers[1].setSection("Delta Squad");

        // Alpha Squad Member

        managerOneStaffs[0] = new Staff();
        managerOneStaffs[0].setName("Bravo");
        managerOneStaffs[0].setNip("230002");
        managerOneStaffs[0].setBracket("2");
        managerOneStaffs[0].setOvertime(10);
        managerOneStaffs[0].setOvertimePay(10_000);

        managerOneStaffs[1] = new Staff();
        managerOneStaffs[1].setName("Charlie");
        managerOneStaffs[1].setNip("230003");
        managerOneStaffs[1].setBracket("2");
        managerOneStaffs[1].setOvertime(10);
        managerOneStaffs[1].setOvertimePay(55_000);

        managers[0].setStaffs(managerOneStaffs);

        // Delta Squad Member

        managerTwoStaffs[0] = new Staff();
        managerTwoStaffs[0].setName("Echo");
        managerTwoStaffs[0].setNip("230005");
        managerTwoStaffs[0].setBracket("3");
        managerTwoStaffs[0].setOvertime(15);
        managerTwoStaffs[0].setOvertimePay(5_500);

        managerTwoStaffs[1] = new Staff();
        managerTwoStaffs[1].setName("Foxtrot");
        managerTwoStaffs[1].setNip("230006");
        managerTwoStaffs[1].setBracket("4");
        managerTwoStaffs[1].setOvertime(5);
        managerTwoStaffs[1].setOvertimePay(100_000);

        managerTwoStaffs[2] = new Staff();
        managerTwoStaffs[2].setName("Golf");
        managerTwoStaffs[2].setNip("230007");
        managerTwoStaffs[2].setBracket("3");
        managerTwoStaffs[2].setOvertime(6);
        managerTwoStaffs[2].setOvertimePay(20_000);

        managers[1].setStaffs(managerTwoStaffs);

        // Managers Info

        managers[0].showInfo();
        managers[1].showInfo();
    }
}

class Employee {
    private String name;
    private String nip;
    private String bracket;
    private double salary;

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

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

    public void setBracket(String bracket) {
        this.bracket = bracket;

        switch (bracket.charAt(0)) {
            case '1':
                this.salary = 5_000_000;
                break;
            case '2':
                this.salary = 3_000_000;
                break;
            case '3':
                this.salary = 2_000_000;
                break;
            case '4':
                this.salary = 1_000_000;
                break;
            case '5':
                this.salary = 750_000;
                break;
        }
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    public String getName() {
        return name;
    }

    public String getNip() {
        return nip;
    }

    public String getBracket() {
        return bracket;
    }

    public double getSalary() {
        return salary;
    }
}

class Staff extends Employee {
    private int overtime;
    private double overtimePay;

    public void setOvertime(int overtime) {
        this.overtime = overtime;
    }

    public int getOvertime() {
        return overtime;
    }

    public void setOvertimePay(double overtimePay) {
        this.overtimePay = overtimePay;
    }

    public double getOvertimePay() {
        return overtimePay;
    }

    public double getSalary(int overtime, double overtimePay) {
        return super.getSalary() + overtime * overtimePay;
    }

    @Override
    public double getSalary() {
        return super.getSalary() + overtime * overtimePay;
    }

    public void showInfo() {
        System.out.printf("NIP            : %s%n", super.getNip());
        System.out.printf("Name           : %s%n", super.getName());
        System.out.printf("Bracket        : %s%n", super.getBracket());
        System.out.printf("Overtime       : %,d%n", this.getOvertime());
        System.out.printf("Overtime Pay : %,.0f%n",
         ↪  this.getOvertimePay());
        System.out.printf("Salary       : %,.0f%n", this.getSalary());
    }
}

class Manager extends Employee {
    private double bonus;
    private String section;
    private Staff[] staffs;

    public void setBonus(double bonus) {
        this.bonus = bonus;
    }

    public double getBonus() {
        return bonus;
    }

    public void setSection(String section) {
        this.section = section;
    }

    public String getSection() {
        return section;
    }

    public void setStaffs(Staff[] staffs) {
        this.staffs = staffs;
    }

    public Staff[] getStaffs() {
        return staffs;
    }

    public void viewStaff() {
        System.out.println("--------------------------------");
        for (Staff staff : staffs) {
            staff.showInfo();
            System.out.println("--------------------------------");
        }
    }
```

```java

    public void showInfo() {
        System.out.printf("Manager of   : %s%n", this.getSection());
        System.out.printf("NIP          : %s%n", this.getNip());
        System.out.printf("Name         : %s%n", this.getName());
        System.out.printf("Bracket      : %s%n", this.getBracket());
        System.out.printf("Bonus        : %,.0f%n", this.getBonus());
        System.out.printf("Salary       : %,.0f%n", this.getSalary());
        System.out.printf("Section      : %s%n", this.getSection());
        this.viewStaff();
    }

    @Override
    public double getSalary() {
        return super.getSalary() + bonus;
    }
}
```

Terminal

```
D:\Kuliah\Smt 3\Object Oriented Programming\Week
↪  10\Polymorphism\src\experiment1>java Main.java
Manager of    : Alpha Squad
NIP           : 230001
Name          : Alpha
Bracket       : 1
Bonus         : 5,000,000
Salary        : 10,000,000
Section       : Alpha Squad
--------------------------------
NIP           : 230002
Name          : Bravo
Bracket       : 2
Overtime      : 10
Overtime Pay  : 10,000
Salary        : 3,100,000
--------------------------------
NIP           : 230003
Name          : Charlie
Bracket       : 2
Overtime      : 10
Overtime Pay  : 55,000
Salary        : 3,550,000
```

```
--------------------------------
Manager of   : Delta Squad
NIP          : 230004
Name         : Delta
Bracket      : 1
Bonus        : 2,500,000
Salary       : 7,500,000
Section      : Delta Squad
--------------------------------
NIP          : 230005
Name         : Echo
Bracket      : 3
Overtime     : 15
Overtime Pay : 5,500
Salary       : 2,082,500
--------------------------------
NIP          : 230006
Name         : Foxtrot
Bracket      : 4
Overtime     : 5
Overtime Pay : 100,000
Salary       : 1,500,000
--------------------------------
NIP          : 230007
Name         : Golf
Bracket      : 3
Overtime     : 6
Overtime Pay : 20,000
Salary       : 2,120,000
--------------------------------
```

2 Exercise

```java
1  public class PerkalianKu {
2      void perkalian(int a, int b) {
3          System.out.println(a * b);
4      }
5
6      void perkalian(int a, int b, int c) {
7          System.out.println(a * b * c);
8      }
9
10     public static void main(String[] args) {
11         PerkalianKu objek = new PerkalianKu();
12
13         objek.perkalian(25, 43);
14         objek.perkalian(34, 23, 56);
15     }
16 }
```

2.1 Dari source coding diatas terletak dimanakah overloading?

`Answer:`
line 2 and line 6 declared the method overloading.

2.2 Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

`Answer:`
both method have a diferent amount of parameter with the first having 2 and the second having 3.

```java
1  public class PerkalianKu {
2      void perkalian(int a, int b) {
3          System.out.println(a * b);
4      }
5
6      void perkalian(double a, double b) {
7          System.out.println(a * b);
8      }
9
10     public static void main(String[] args) {
11         PerkalianKu objek = new PerkalianKu();
12
13         objek.perkalian(25, 43);
14         objek.perkalian(34.56, 23.7);
15     }
16 }
```

2.3 Dari source coding diatas terletak dimanakah overloading?

`Answer:`
line 2 and line 6 declared the method overloading.

2.4 Jika terdapat overloading ada berapa tipe parameter yang berbeda?

`Answer:`
it is a diferent data type for each parameter.

```java
1  class Ikan {
2      public void swim() {
3          System.out.println("Ikan bisa berenang");
4      }
5  }
6  class Piranha extends Ikan {
7      public void swim() {
8          System.out.println("Piranha bisa makan daging");
9      }
10 }
11 public class Fish {
12     Ikan a = new Ikan();
13     Ikan b = new Piranha();
14     a.swim();
15     b.swim();
16 }
```

2.5 Dari source coding diatas terletak dimanakah overriding?

`Answer:`
line 7 override the method from its superclass in line 2.

2.6 Jabarkanlah apabila sourcoding diatas jika terdapat overriding?

`Answer:`
yes there is a method override in the source code. The extended Piranha class override the swim method it inherited from the Ikan superclass.

3 Assignment

3.1 Overloading

```java
package assignment;

public class Triangle {
    private int angle;

    public int sumAngle(int angleA) {
        return 180 - angleA;
    }

    public int sumAngle(int angleA, int angleB) {
        return 180 - angleA - angleB;
    }

    public int circumference(int sideA, int sideB, int sideC) {
        return sideA + sideB + sideC;
    }

    public double circumference(int sideA, int sideB) {
        return sideA + sideB + Math.sqrt((sideA * sideA) + (sideB *
         ↪  sideB));
    }

    public void setAngle(int angle) {
        this.angle = angle;
    }

    public int getAngle() {
        return angle;
    }
}
```

3.2 Overriding

```
package assignment;

public class Human {
    public void breath() {
        System.out.println("breath");
    }

    public void eat() {
        System.out.println("Om Nom Nom");
    }
}

class Lecturer extends Human {
    @Override
    public void eat() {
        System.out.println("Slurp");
    }

    public void overtime() {
        System.out.println("Doing overtime just like a horse");
    }
}

class Student extends Human {
    @Override
    public void eat() {
        System.out.println("Chomp Chomp Chomp");
    }

    public void sleep() {
        System.out.println("ZZZ - Atarashii Gakko No Leaders");
    }
}
```