Object Oriented Programming
Inheritance

Name
Muhammad Baihaqi Aulia Asy'ari

NIM
2241720145

Class
2I

Department
Information Technology

Study Program
D4 Informatics Engineering

1 Experiment 1

Main.java

```
package experiment1;

public class Main {
    public static void main(String[] args) {
        Manager manager = new Manager();
        manager.name = "Anu";
        manager.address = "Home";
        manager.age = 101;
        manager.gender = "Fe male";
        manager.salary = 3_000_000;
        manager.bonus = 1_000_000;
        manager.showManagerData();

        Staff staff = new Staff();
        staff.name = "Itu";
        staff.address = "Alone";
        staff.age = 42;
        staff.gender = "Fe male";
        staff.salary = 2_000_000;
        staff.overtime = 500_000;
        staff.paycut = 250_000;
        staff.showStaffData();
    }
}

class Employee {
    public String name, address, gender;
    public int age, salary;

    public Employee() {
    }

    public Employee(String name, String address, String gender, int
     ↪  age, int salary) {
        this.name = name;
        this.address = address;
        this.gender = gender;
        this.age = age;
        this.salary = salary;
    }

    public void showEmployeeData() {
        System.out.println(String.format("Name          : %s", name));
        System.out.println(String.format("Address       : %s",
        ↪   address));
        System.out.println(String.format("Gender        : %s",
        ↪   gender));
        System.out.println(String.format("Age           : %d", age));
        System.out.println(String.format("Salary        : %,d",
        ↪   salary));
    }
}

class Manager extends Employee {
    public int bonus;

    public Manager() {
    }

    public void showManagerData() {
        super.showEmployeeData();
        System.out.println(String.format("Bonus         : %,d",
        ↪   bonus));
        System.out.println(String.format("Total Salary  : %,d",
        ↪   super.salary+bonus));
    }
}

class Staff extends Employee {
    public int overtime, paycut;

    public Staff() {
    }

    public Staff(String name, String address, String gender, int age,
    ↪   int salary, int overtime, int paycut) {
        super(name, address, gender, age, salary);
        this.overtime = overtime;
        this.paycut = paycut;
    }

```

```java
    public void showStaffData() {
        super.showEmployeeData();
        System.out.println(String.format("Overtime      : %,d",
         ↪  overtime));
        System.out.println(String.format("Paycut        : %,d",
         ↪  paycut));
        System.out.println(String.format("Total Salary  : %,d",
         ↪  super.salary+overtime-paycut));
    }
}
```

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' 'experiment1.Main'
Name          : Anu
Address       : Home
Gender        : Fe male
Age           : 101
Salary        : 3,000,000
Bonus         : 1,000,000
Total Salary  : 4,000,000
Name          : Itu
Address       : Alone
Gender        : Fe male
Age           : 42
Salary        : 2,000,000
Overtime      : 500,000
Paycut        : 250,000
Total Salary  : 2,250,000
```

1.1 Question

1. Sebutkan class mana yang termasuk super class dan sub class dari percobaan 1 diatas!

2. Kata kunci apakah yang digunakan untuk menurunkan suatu class ke class yang lain?

3. Perhatikan kode program pada class Manager, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Karyawan!

4. Jelaskan kata kunci super pada potongan program dibawah ini yang terdapat pada class Manager!

   ```
   System.out.println(String.format("Total Salary  : %,d",
   ↪  super.salary+bonus));
   ```

5. Program pada percobaan 1 diatas termasuk dalam jenis inheritance apa? Jelaskan alasannya!

1.2 Answer

1. The Employee class is a superclass and The Manager and Staff class are the subclass.

2. extends

3. name, address, gender, age, and salary are inherited from the Employee class. The bonus attribute is the only attribute that is not inherited from the Employee class.

4. The super keyword refer to the superclass attributes or methods. In this case it is used to refer to the salary attribute of the superclass.

5. It is a Hierarchical Inheritance because the superclass has more than one subclass.

2 Experiment 2

Main.java

```java
package experiment2;

public class Main {
    public static void main(String[] args) {
        PermanentStaff permanentStaff = new PermanentStaff("Anu", "Home", "Fe Male", 34, 2_000_000, 250_000, 200_000, "2A", 100_000);
        permanentStaff.showPermanentStaffData();

        DailyStaff dailyStaff = new DailyStaff("Itu", "Alone", "Fe Male", 1738, 10_000, 100_000, 50_000, 100);
        dailyStaff.showDailyStaffData();
    }
}

class Employee {
    public String name, address, gender;
    public int age, salary;

    public Employee() {
    }

    public Employee(String name, String address, String gender, int age, int salary) {
        this.name = name;
        this.address = address;
        this.gender = gender;
        this.age = age;
        this.salary = salary;
    }

    public void showEmployeeData() {
        System.out.println(String.format("Name              : %s", name));
        System.out.println(String.format("Address           : %s", address));
        System.out.println(String.format("Gender            : %s", gender));
        System.out.println(String.format("Age               : %d", age));
        System.out.println(String.format("Salary            : %,d", salary));
    }
}

class Manager extends Employee {
    public int bonus;

    public Manager() {
    }

    public void showManagerData() {
        super.showEmployeeData();
        System.out.println(String.format("Bonus             : %,d", bonus));
        System.out.println(String.format("Total Salary      : %,d", super.salary+bonus));
    }
}

class Staff extends Employee {
    public int overtime, paycut;

    public Staff() {
    }

    public Staff(String name, String address, String gender, int age, int salary, int overtime, int paycut) {
        super(name, address, gender, age, salary);
        this.overtime = overtime;
        this.paycut = paycut;
    }

    public void showStaffData() {
        super.showEmployeeData();
        System.out.println(String.format("Overtime          : %,d", overtime));
        System.out.println(String.format("Paycut            : %,d", paycut));
        System.out.println(String.format("Total Salary      : %,d", super.salary+overtime-paycut));
    }
}

class PermanentStaff extends Staff {
    public String bracket;
    public int insurance;

    public PermanentStaff() {
    }

    public PermanentStaff(String name, String address, String gender, int age, int salary, int overtime, int paycut, String bracket, int insurance) {
        super(name, address, gender, age, salary, overtime, paycut);
        this.bracket = bracket;
        this.insurance = insurance;
    }

    public void showPermanentStaffData() {
        String bar = "================================";
        String template = String.format("%%%ds%%-%ds%%%ds", 16,"Permanent Staff Data".length(), 16);
        System.out.println(String.format(template, bar,"Permanent Staff Data", bar));
        super.showStaffData();
        System.out.println(String.format("Bracket           : %s", bracket));
        System.out.println(String.format("Insurance         : %,d", insurance));
        System.out.println(String.format("Net Salary        : %,d", super.salary+overtime-paycut-insurance));
    }
}

class DailyStaff extends Staff {
    public int totalWorkHours;

    public DailyStaff() {
    }

    public DailyStaff(String name, String address, String gender, int age, int salary, int overtime, int paycut, int totalWorkHours) {
        super(name, address, gender, age, salary, overtime, paycut);
        this.totalWorkHours = totalWorkHours;
    }
    public void showDailyStaffData() {
        String bar = "================================";
        String template = String.format("%%%ds%%-%ds%%%ds", 16,"Daily Staff Data".length(), 16);
        System.out.println(String.format(template, bar,"Daily Staff Data", bar));
        super.showStaffData();
        System.out.println(String.format("Total Work Hours  : %d", totalWorkHours));
        System.out.println(String.format("Net Salary        : %,d", super.salary*totalWorkHours+overtime-paycut));
    }
}
```

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\80d97a47d24665dc0bce
 ↪  'experiment2.Main'
================================Permanent Staff
 ↪  Data================================
Name              : Anu
Address           : Home
Gender            : Fe Male
Age               : 34
Salary            : 2,000,000
Overtime          : 250,000
Paycut            : 200,000
Total Salary      : 2,050,000
Bracket           : 2A
Insurance         : 100,000
Net Salary        : 1,950,000
================================Daily Staff
 ↪  Data================================
Name              : Itu
Address           : Alone
Gender            : Fe Male
Age               : 1738
Salary            : 10,000
Overtime          : 100,000
Paycut            : 50,000
Total Salary      : 60,000
Total Work Hours  : 100
Net Salary        : 1,050,000
```

2.1 Question

1. Berdasarkan class diatas manakah yang termasuk single inheritance dan mana yang termasuk multilevel inheritance?

2. Perhatikan kode program class StaffTetap dan StaffHarian, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Staff!

3. Apakah fungsi potongan program berikut pada class StaffHarian

   ```
   super(name, address, gender, age, salary, overtime, paycut);
   ```

4. Apakah fungsi potongan program berikut pada class StaffHarian

   ```
   super.showStaffData();
   ```

5. Perhatikan kode program dibawah ini yang terdapat pada class StaffTetap

   ```
   System.out.println(String.format("Net Salary        : %,d",
    ↪  super.salary+overtime-paycut-insurance));
   ```

   Terlihat dipotongan program diatas atribut gaji, lembur dan potongan dapat diakses langsung. Kenapa hal ini bisa terjadi dan bagaimana class StaffTetap memiliki atribut gaji, lembur, dan potongan padahal dalam class tersebut tidak dideklarasikan atribut gaji, lembur, dan potongan?

2.2 Answer

1. Manager and Staff class are single inheritance, while DailyStaff and permanentStaff are multilevel inheritance.

2. Both class inherited the name, address, gender, age, salary, overtime, paycut from the Employee class. The DailyStaff has the attribute totalWorkHours of their own and the permanentStaff has the attributes bracket, and insurance.

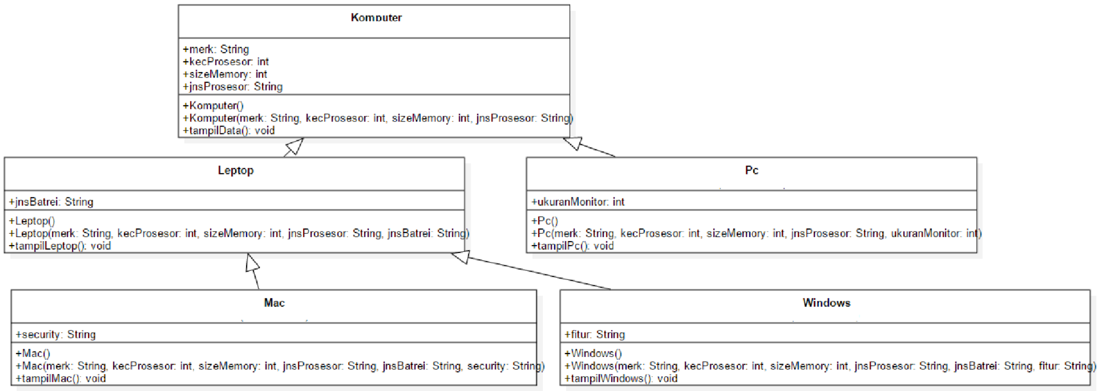
3. It is used to construct the superclass with the required attributes of the superclass from the subclass constructor parameter.

4. It is used to call a method from the superclass.

5. Because the subclass also inherit the attribute and the attribute it self is public thus making it accessable to the subclass.

3 Assignment

Buatlah sebuah program dengan konsep pewarisan seperti pada class diagram berikut ini. Kemudian buatlah instansiasi objek untuk menampilkan data pada class Mac, Windows dan Pc!.


Komputer
+merk: String
+kecProsesor: int
+sizeMemory: int
+jnsProsesor: String
+Komputer()
+Komputer(merk: String, kecProsesor: int, sizeMemory: int, jnsProsesor: String)
+tampilData(): void
Leptop
+jnsBatrei: String
+Leptop()
+Leptop(merk: String, kecProsesor: int, sizeMemory: int, jnsProsesor: String, jnsBatrei: String)
+tampilLeptop(): void
Pc
+ukuranMonitor: int
+Pc()
+Pc(merk: String, kecProsesor: int, sizeMemory: int, jnsProsesor: String, ukuranMonitor: int)
+tampilPc(): void
Mac
+security: String
+Mac()
+Mac(merk: String, kecProsesor: int, sizeMemory: int, jnsProsesor: String, jnsBatrei: String, security: String)
+tampilMac(): void
Windows
+fitur: String
+Windows()
+Windows(merk: String, kecProsesor: int, sizeMemory: int, jnsProsesor: String, jnsBatrei: String, fitur: String)
+tampilWindows(): void

Main.java

```java
package assignment;

public class Main {
    public static void main(String[] args) {
        PC pc = new PC("Dell", 3_600, 16, "AMD Ryzen 5 3500x", 24);
        pc.showPC();
        Mac mac = new Mac("Apple", 3_200, 8, "M1", "udisclosed information", "XProtect");
        mac.showMac();
        Windows windows = new Windows("ROG Strix G15 G513RM", 3_200, 8, "AMD Ryzen 7 6800H", "56WHrs", "NVIDIA GeForce RTX 3060 Laptop GPU");
        windows.showWindows();
    }
}

class Computer {
    public String brand;
    public int coreClock;
    public int ramSize;
    public String processorName;

    public Computer() {
    }

    public Computer(String brand, int coreClock, int ramSize, String processorName) {
        this.brand = brand;
        this.coreClock = coreClock;
        this.ramSize = ramSize;
        this.processorName = processorName;
    }

    public void showData() {
        System.out.println(String.format("Brand         : %s", brand));
        System.out.println(String.format("Core Clock    : %,d Mhz", coreClock));
        System.out.println(String.format("RAM           : %d Gb", ramSize));
        System.out.println(String.format("Processor Name: %s", processorName));
    }
}

class PC extends Computer {
    public int monitorSize;

    public PC() {
    }

    public PC(String brand, int coreClock, int ramSize, String processorName, int monitorSize) {
        super(brand, coreClock, ramSize, processorName);
        this.monitorSize = monitorSize;
    }

    public void showPC() {
        super.showData();
        System.out.println(String.format("Monitor Size  : %d inch", monitorSize));
    }
}

class Laptop extends Computer {
    public String batteryType;
```

```java

    public Laptop() {
    }

    public Laptop(String brand, int coreClock, int ramSize, String processorName, String batteryType) {
        super(brand, coreClock, ramSize, processorName);
        this.batteryType = batteryType;
    }

    public void showLaptop() {
        super.showData();
        System.out.println(String.format("Battery Type  : %s", batteryType));
    }
}

class Mac extends Laptop {
    public String security;

    public Mac() {
    }

    public Mac(String brand, int coreClock, int ramSize, String processorName, String batteryType, String security) {
        super(brand, coreClock, ramSize, processorName, batteryType);
        this.security = security;
    }

    public void showMac() {
        super.showLaptop();
        System.out.println(String.format("Security      : %s", security));
    }
}

class Windows extends Laptop {
    public String features;

    public Windows() {
    }

    public Windows(String brand, int coreClock, int ramSize, String
     ↪  processorName, String batteryType, String features) {
        super(brand, coreClock, ramSize, processorName, batteryType);
        this.features = features;
    }

    public void showWindows() {
        super.showLaptop();
        System.out.println(String.format("Features      : %s",
         ↪  features));
    }
}
```

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' 'assignment.Main'
Brand         : Dell
Core Clock    : 3,600 Mhz
RAM           : 16 Gb
Processor Name: AMD Ryzen 5 3500x
Monitor Size  : 24 inch

Brand         : Apple
Core Clock    : 3,200 Mhz
RAM           : 8 Gb
Processor Name: M1
Battery Type  : udisclosed information
Security      : XProtect

Brand         : ROG Strix G15 G513RM
Core Clock    : 3,200 Mhz
RAM           : 8 Gb
Processor Name: AMD Ryzen 7 6800H
Battery Type  : 56WHrs
Features      : NVIDIA GeForce RTX 3060 Laptop GPU
```