

1. Below, the class `Manager` inherits from `Employee`, and `equals` has been overridden in `Employee` using the instanceof strategy. Your development team decides `Manager` needs to have its own `equals` method that takes into account the `bonus` field. Your team decides to proceed by using composition instead of inheritance. Rewrite the code shown below so that composition instead of inheritance is used and each class has its own `equals` method (and the `bonus` field is used in determining equality between two `Manager` objects).

NOTE: You must *completely rewrite* the `Manager` class. You do not need to modify `Employee`.

```
class Employee {
    private String name;
    private int salary;
    private LocalDate hireDay;

    Employee(String aName, int aSalary, int aYear,
              int aMonth, int aDay) {
        name = aName;
        salary = aSalary;
        hireDay = LocalDate.of(aYear, aMonth, aDay);
    }

    public String getName() {
        return name;
    }

    public LocalDate getHireDay() {
        return hireDay;
    }

    public int getSalary() {
        return salary;
    }

    @Override
    public final boolean equals(Object ob) {
        if (ob == null) return false;
        if (!(ob instanceof Employee)) return false;
        Employee e = (Employee) ob;
        return (e.name.equals(name) && e.salary==salary
                && e.hireDay.equals(hireDay));
    }
}

class Manager extends Employee {
    private int bonus;
```

```

    public Manager(String name, int salary,
        int year, int month, int day) {
        super(name, salary, year, month, day);
        bonus = 0;
    }

    @Override
    public int getSalary() {
        int baseSalary = super.getSalary();
        return baseSalary + bonus;
    }

    public void setBonus(int b) {
        bonus = b;
    }
}

```

## Solution:

No change to Employee class. New Manager class below:

```

class Manager {
    private Employee e;
    public Manager(String name, int salary,
        int year, int month, int day) {
        e = new Employee(name, salary, year, month, day);
        bonus = 0;
    }

    public String getName() {
        return e.getName();
    }

    public LocalDate getHireDay() {
        return e.getHireDay();
    }

    public int getSalary() {
        int baseSalary = e.getSalary();
        return baseSalary + bonus;
    }
}

```

```
public void setBonus(int b) {  
    bonus = b;  
}
```

```
@Override
```

```
public boolean equals(Object ob) {  
    if(ob == null) return false;  
    if(getClass() != ob.getClass()) return false;  
    Manager m = (Manager)ob;  
    return m.e.equals(e) && m.bonus == bonus;  
}
```

```
private int bonus;
```

```
}
```