```
/**
 * Assignment for your lecture 2. Please finish all the questions
under 'Assignment'
 * Please try to think the extra credit question.
 * The deadline of this assignment is 09/21/2018 23:59 PST.
 * Please feel free to contact Amanda and Zane for any questions.
 */

class Employee {
    String name;
    int age;
    Gender gender;
    double salary;// salary per month

    // Constructor. Please set all the data in constructor.
    public Employee(String name, int age, Gender gender, double
salary) {
                //write your code here
        this.name = name;
        this.age = age;
        this.gender = gender;
        this.salary = salary;
    }

    // Getter for `name`. Return the current `name` data
    public String getName() {
                //write your code here
        return this.name;
    }

    // Setter for `name`. Set `name` data
    public void setName(String name) {
                //write your code here
        this.name = name;
    }
}

enum Gender {
    MALE,
    FEMALE;
}


public class Assignment2 {
    // Assignment

    /**
     * Write a method to calculate the Social Security Tax of an
employee and print it.
     * If the salary is less than or equal to 8900, the Social
```

```
Security Tax is 6.2% of the salary.
     * If the salary is more than 8900, the Social Security Tax is
6.2% of 106,800.
     */
    public double socialSecurityTax(Employee employee) {
        //write your code here
        if(employee.salary <= 8900) {
                System.out.println(String.format("the social Security
Tax of %s is: %s", employee.name, employee.salary*0.062));
                return employee.salary*0.062;
        } else {
                System.out.println(String.format("the social Security
Tax of %s is: %s", employee.name,(double)(106800*0.062));
                return (double)(106800*0.062);
        }
    }

    /**
     * Write a method to calculate an employee's contribution for
insurance coverage and print it.
     * Amount of deduction is computed as follows:
     * If the employee is under 35, rate is 3% of salary; if the
employee is between 35 and 50(inclusive), rate is 4% of salary;
     * If the employee is between 50 and 60(exclusive), rate is 5% of
salary; If the employee is above 60, rate is 6% of salary.
     */
    public double insuranceCoverage(Employee employee) {
        //write your code here
        if(employee.age < 35) {
                System.out.println(String.format("the employee %s's
contribution for insurance coverage is: %s", employee.name,
employee.salary * 0.03);
                return  employee.salary * 0.03;
        } else if(employee.age >= 35 && employee.age <= 50) {
                System.out.println(String.format("the employee %s's
contribution for insurance coverage is: %s", employee.name,
employee.salary * 0.04);
                return  employee.salary * 0.04;
        } else if(employee.age > 50 && employee.age < 60) {
                System.out.println(String.format("the employee %s's
contribution for insurance coverage is: %s", employee.name,
employee.salary * 0.05);
                return  employee.salary * 0.05;
        } else {
                System.out.println(String.format("the employee %s's
contribution for insurance coverage is: %s", employee.name,
employee.salary * 0.06);
                return  employee.salary * 0.06;
        }
    }
```

```
    /**
     * Write a method to sort three employees' salary from low to
high, and then print their name in order.
     * For example, Alice's salary is 1000, John's salary is 500,
Jenny's salary is 1200, you should print:
     * John Alice Jenny
     */
    public void sortSalary(Employee e1, Employee e2, Employee e3) {
        //write your code here
        if(e1.salary < e2.salary && e1.salary < e3.salary) {
                if(e2.salary < e3.salary) {
                        System.out.println(e1.name + " " + e2.name +
" " + e3.name);
                } else {
                        System.out.println(e1.name + " " + e3.name +
" " + e2.name);
                }
        }
        else if(e2.salary < e1.salary && e2.salary < e3.salary) {
                if(e1.salary < e3.salary) {
                        System.out.println(e2.name + " " + e1.name +
" " + e3.name);
                } else {
                        System.out.println(e2.name + " " + e3.name +
" " + e1.name);
                }
        }
        else if(e3.salary < e1.salary && e3.salary < e2.salary) {
                if(e2.salary < e1.salary) {
                        System.out.println(e3.name + " " + e2.name +
" " + e1.name);
                } else {
                        System.out.println(e3.name + " " + e1.name +
" " + e2.name);
                }
        }
    }

    /**
     * Write a method to raise an employee's salary to three times of
his/her original salary.
     * Eg: original salary was 1000/month. After using this method,
the salary is 3000/month.
     * Do not change the input of this method.
     * Try to add a new method in Employee class: public void
raiseSalary(double byPercent)
     */
    public void tripleSalary(Employee employee) {
        //write your code here
```

```
        employee.salary = (double)(3*employee.salary);
    }

    public void raiseSalary(double byPercent) {
        this.salary = (double)(this.salary * (1+byPercent));
    }


    //Extra credit

    /**
     * I have written some code below. What I want is to swap two
Employee objects.
     * One is Jenny and one is John. But after running it, I got the
result below:
     * Before: a=Jenny
     * Before: b=John
     * After: a=Jenny
     * After: b=John
     * There is no change after swap()! Do you know the reason why my
swap failed?
     * Write your understanding of the reason and explain it.
     */
        the first two lines, we declare two employee instances which
the first employee named Jenny, the second one
        named John. Variable a is the reference of employee Jenny.
Variable b is the reference of employee John.
        When we call function swap(), we create two new references x
and y that x points to the employee Jenny which
        reference a points to and y points to the employee Jonh which
reference b points to. So when we swap x and y,
        we swap reference x and reference y which means x points to
employee John, y points to employee Jenny. It dose
        not impact reference a and reference b. So reference a still
points to employee Jenny and reference b still points to
        employee John.
    /*
     write your understanding here.
    */
    public static void main(String[] args) {
        Employee a = new Employee("Jenny", 20, Gender.FEMALE, 2000);
        Employee b = new Employee("John", 30, Gender.MALE, 2500);
        System.out.println("Before: a=" + a.getName());
        System.out.println("Before: b=" + b.getName());
        swap(a, b);
        System.out.println("After: a=" + a.getName());
        System.out.println("After: b=" + b.getName());
    }

    public static void swap(Employee x, Employee y) {
```

```
        Employee temp = x;
        x = y;
        y = temp;
    }
}
```