Credit Name: CSE 3130 - Object Oriented Programming 2

Assignment: UEmployee

How has your program changed from planning to coding to now? Please Explain

```
public class UEmployee {
    private String fName;
    private String lName;
    private double pay;

    public UEmployee(String fname, String lname, double sal)
    {
        fName = fname;
        lName = lname;
        pay = sal;
    }
```

Our superclass is the UEmployee class.
I created the class variables which include the first name, last name and their pay/salary.
I then formed a constructor and initialized each of the class variables with the parameters of the class method.

```
public String getName()
{
    return (fName + " " + lName);
}
```

I created a method to get the name of the employee by concatenating the first and last name together.

```
public double getSalary()
{
    return pay;
}
```

Then there is a method which simply returns the salary of the employee.

```java
public class Faculty extends UEmployee
{
    private String dep_name;

    public Faculty(String dep, String fname,String lname, double sal)
    {
        super(fname,lname,sal);
        dep_name = dep;

    }
```

Our first subclass Faculty extends the UEmployee class and has an extra class variable which indicates the name of the department. As such, in our constructor we call and create the superclass as an object, with our parameters and also initialize our department name using the respective parameter.

```java
public String getDepName()
{
    return dep_name;
}
```

The only additional method we need is to return the department name

```java
public class Staff extends UEmployee
{
    private String job_name;

    public Staff(String job, String fname, String lname, double sal)
    {
        super(fname,lname,sal);

        job_name = job;
    }
```

Our last subclass is the staff class which also extends the UEmployee, which requires an extra variable for the job name of the employee. As such, in our constructor we call and create the superclass as an object, with our parameters and also initialize our jobname using the respective parameter.

```java
public String getJobName()
{
    return job_name;
}
```

We create a method which returns the job name.

```java
public class University
{

    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        int department = 0;
        int title = 0;
        String dep_name = null;
        String job_name = null;
        double salary = 0;
        double pay = 0 ;
        int selection = -1;



            System.out.println("\n============= University =============");

            System.out.println("Do you want to Register a Staff member: (Y)es or (N)o");

            String createAcct = null;
            do
            {
                createAcct = input.next().toUpperCase();


                if(!(createAcct.equals("Y")) && !(createAcct.equals("N")))
                {
                    System.out.println("Invalid Input, Try Again.");
                }

            }
            while(!(createAcct.equals("Y")) && !(createAcct.equals("N")));
```

Our tester class is University, and is set up very similarly to the bank class.

The way i interpreted the assignment was that both the faculty and staff classes are both specific to one person and so the salary in the faculty class is a base salary and the pay in the staff class acts as a multiplier based on your position.

Similar to the bank assignment, we ask the user if he would like to register a staff member and encase their response in a do while loop to ensure a valid response and

catch errors. This loop continues as long as neither "y" nor "n" is registered for the createAcct variable.

```
if(createAcct.equals("N"))
{
    System.out.println("Quitting the University Application");
    System.exit(0);;
}
```

if they selected "n" the application closes.

```
//Get Name
System.out.print("Enter your First Name: ");
String fname = input.next();

System.out.print("Enter your Last Name: ");
String lname = input.next();
```

Then we get their first and last name from them.

```
do {

    System.out.println("Enter the faculty to be joined:");
    System.out.println("\n(1) Faculty of Arts\n(2) Faculty of Science\n(3) Faculty of Business");
    department = input.nextInt();


    switch(department)
    {
    case 1: dep_name = "Arts"; salary = 20 ; break;

    case 2: dep_name = "Science"; salary = 25; break;

    case 3: dep_name = "Business"; salary = 30; break;

    default: System.out.println("Error. Incorrect Input.");
    }

}while(!(department == 1) && !(department == 2) && !(department == 3));
```

Then we ask them which faculty they are joining, and we use a switch statement to set their department names and set their salary, specific to the department they joined.

Again we encase in a do while loop to catch errors.

```
Faculty fac = new Faculty(dep_name, fname,lname,salary);
```

We then make a faculty object with the information given and set.

```
do {

    System.out.println("Enter the job title:");
    System.out.println("\n(1)Assistant Professor\n(2)Professor\n(3)Dean");
    title = input.nextInt();

    switch(title)
    {
        case 1: job_name = "Assistant Professor"; pay = 1 ; break;

        case 2: job_name = "Professor"; pay = 2; break;

        case 3: job_name = "Dean"; pay = 3; break;

        default: System.out.println("Error. Incorrect Input.");
    }

    }while(!(title == 1) && !(department == 2) && !(department == 3));

Staff position = new Staff(job_name, fname, lname , pay);
```

We then do a similar switch statement after prompting for their position in their faculty and their choice acts as a multiplier for their base salary indicated by the faculty class.

Then we make a staff object with the given information.

```
do {
    System.out.print("\n========== University Staff Account ==========\n");
    System.out.println(position.getName());
    System.out.println("Choose an Option: ");
    System.out.println("(1)Display Position and Faculty \n(2)Display Salary\n(0)Quit");
    selection = input.nextInt();

    switch(selection)
    {
    case 0: System.out.println("Quitting the Application.");
            System.exit(0); break;

    case 1: System.out.println("Position: " + job_name + ", Faculty of " + dep_name);break;

    case 2: System.out.println("Pay: $" + salary*pay + "/hr ");break;

    }
}
while(selection != 0);
```

Lastly, while they do not select 0(quit) we allow them to check their salary and position using a switch statement:

If they select 0, we quit the application.
If they select 1 we display the position, and the faculty in which they hold their position.
And if they select 2, we display their total hourly salary by multiplying the salary and pay(multiplier) together.