Print

iLab 5 of 6: Inheritance

Submit your assignment to the Dropbox located on the silver tab at the top of this page.

(See the Syllabus section "Due Dates for Assignments & Exams" for due dates.)

Remember This!

Connect to the iLab here.

iLAB OVERVIEW

Scenario and Summary

The objective of the lab is to take the UML Class diagram and enhance last week's Employee class by making the following changes:

1. Create a class called Salaried that is derived from Employee.
2. Create a class called Hourly that is also derived from Employee.
3. Override the base class calculatePay() method.
4. Override the displayEmployee() method.

Deliverables

Due this week:

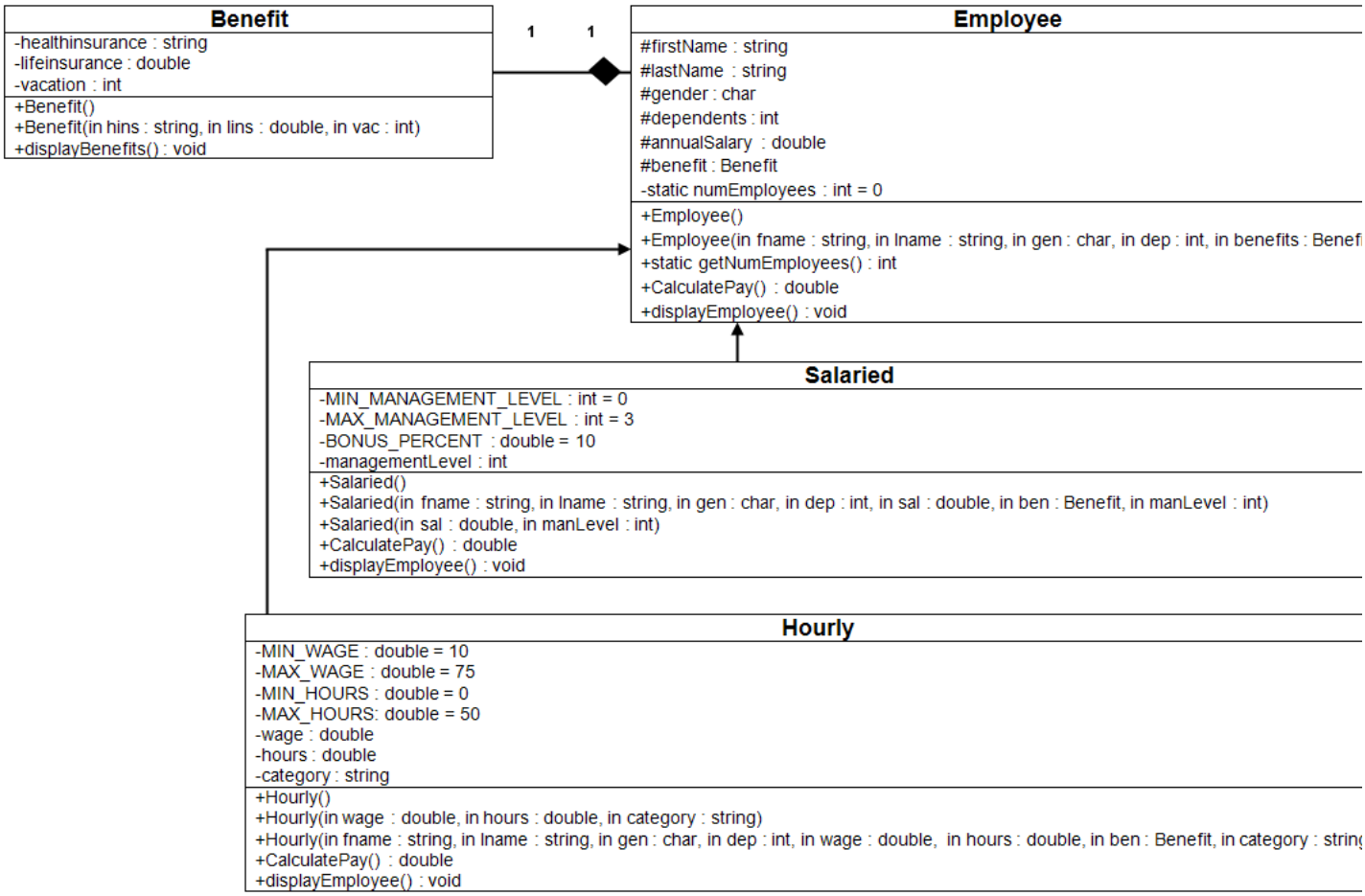
- Capture the Console output window and paste into a Word document.
- Zip the project folder file.
- Put the zip file and screenshots (Word document) in the Dropbox.

iLAB STEPS

STEP 1: Understand the UML Diagram

Back to Top

Notice the change in UML diagram. It is common practice to leave out the accessors and mutators (getters and setters) from UML class diagrams, since there can be so many of them. Unless otherwise specified, it is ***assumed*** that there is an accessor (getter) and a mutator (setter) for every class attribute.



STEP 2: Create the Project

Back to Top

Create a new project and name it CIS247C_WK5_Lab_LASTNAME. Copy all the source files from the Week 4 project into the Week 5 project.

Before you move on to the next step, build and execute the Week 5 project.

STEP 3: Modify the Employee Class

Back to Top

1. Using the updated Employee class diagram, modify the attributes to be protected.
2. Delete the iEmployee interface class, and remove the reference from the Employee class.

STEP 4: Create the Salaried Class

Back to Top

1. Using the UML Diagrams from Step 1, create the Salaried classes, ensuring to specify that the Salary class inherits from the Employee class.
2. For each of the constructors listed in the Salaried class, ensure to invoke the appropriate base class constructor and pass the correct arguments to the base class constructor. This will initialize the protected attributes and update the numEmployees counter.
3. The valid management levels are 0, 1, 2, and 3, and should be implemented as a constant.
4. Override the calculatePay method to add a 10 percent bonus for each of the management levels (i.e., bonus percentage = managementLevel * .10). The bonus percentage should be implemented as a constant.
5. Override the displayEmployee() method to add the management level to the employee information.

STEP 5: Create the Hourly Class

Back to Top

1. Using the UML Diagrams from Step 1, create the Hourly classes, ensuring to specify that the Hourly class inherits from the Employee class.
2. For each of the constructors listed in the Hourly class, ensure to invoke the appropriate base class constructor and pass the correct arguments to the base class constructor. This will initialize the protected attributes and update the numEmployees counter.
3. The valid category types are "temporary", "part time", and "full time".
4. The provided hours must be more than 0 hours and less than 50 hours, and the limits should be implemented as constants.
5. The provided wage must be between 10 and 75, and the limits should be implemented as constants.
6. Override the calculatePay method by multiplying the wages by the number of hours.
7. Override the Employee setAnnualSalary method and set the annual salary by multiplying the weekly pay by 50.

8. Override the displayEmployee() method to add the category to the hourly employee information.

STEP 6: Modify the Main Method

Back to Top

1. Using previous weeks' assignments as an example, create at least one Employee, Hourly, and Salaried employee.
2. For each object created, display the number of employees created.
3. For each object created, write statements to exercise each of the public methods listed in the Class diagram.
4. For each object created, invoke the object's displayEmployee() method to display the employee's information.
5. For employee, the following information needs to be displayed:

Partial Sample output

```
Employee Information
------------------------------------------------------------
Name:           Nana Liu
Gender:         F
Dependents:     2
Annual Salary:  60000.00
Weekly Salary:  1153.85

Benefit Information
------------------------------------------------------------
Health Insurance:       PPO
Life Insurance:         1.50
Vacation:               20 days

--- Number of Employee Object Created ----      Number of employees: 1
```

6. For salaried employee, the following information needs to be displayed:

Partial Sample output

```
Name:           Jackie Chan
Gender:         M
Dependents:     1
Annual Salary:  50000.00
Weekly Salary:  1250.00

Benefit Information
------------------------------------------------------------
Health Insurance:       HMO
Life Insurance:         100.00
Vacation:               16 days
Salaried Employee
Management level:       3
--- Number of Employee Object Created ----      Number of employees: 2
```

7. For hourly employee, the following information needs to be displayed:

Partial Sample output

```
------------------------------------------------------------
Name:           James Bond
Gender:         M
Dependents:     0
Annual Salary:  100000.00
Weekly Salary:  2000.00

Benefit Information
------------------------------------------------------------
Health Insurance:       PPO
Life Insurance:         5.00
Vacation:               17 days
Hourly Employee
Category:               Unknown
Wage:                   40.00
Hours:                  50.00

--- Number of Employee Object Created ----      Number of employees: 3
```

STEP 7: Compile and Test

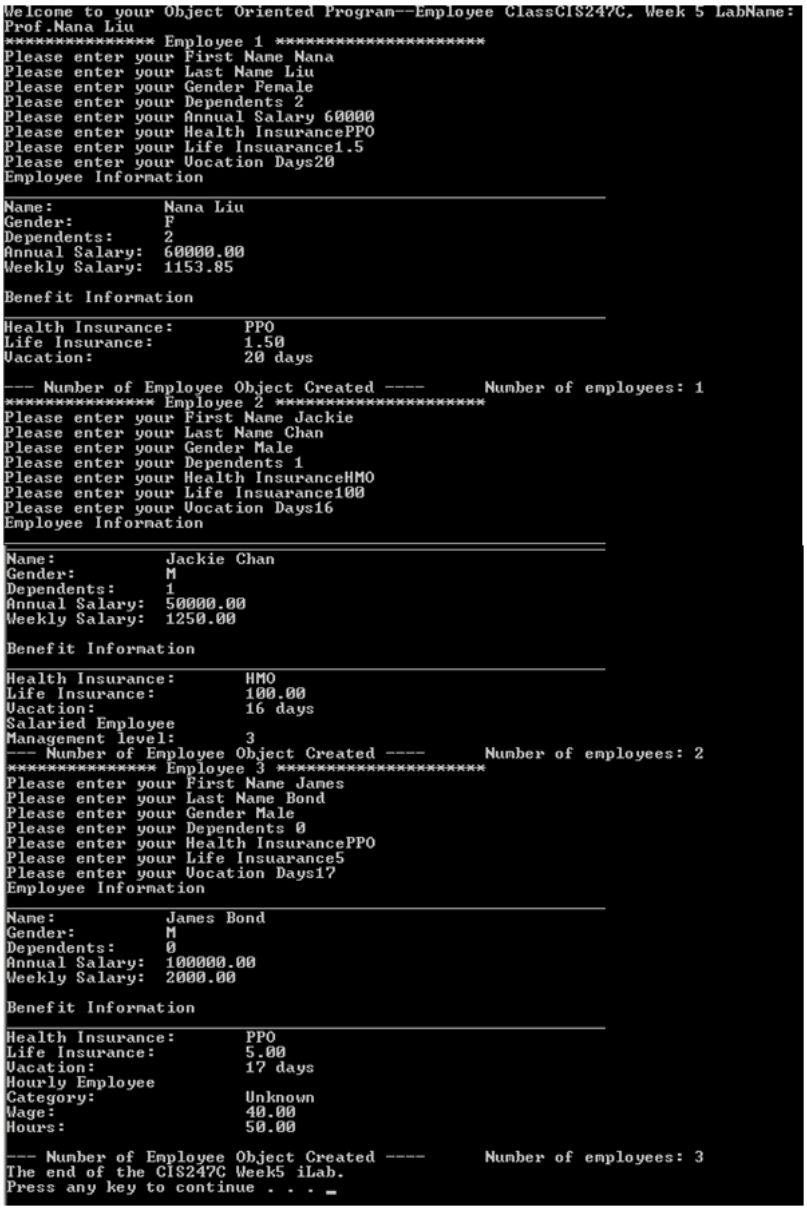
Back to Top

When done, compile and run your code.

Then, debug any errors until your code is error-free.

Check your output to ensure that you have the desired output, modify your code as necessary, and rebuild.

Below is the complete sample program output for your reference.



STEP 8: Label Title

Back to Top

- Capture the Console output window and paste into a Word document.
- Put the zip file and screenshots (Word document) in the Dropbox.

Submit your lab to the Dropbox located on the silver tab at the top of this page. For instructions on how to use the Dropbox, read these step-by-step instructions or watch this Dropbox Tutorial.

See the Syllabus section "Due Dates for Assignments & Exams" for due date information.

Back to Top