Print

iLab 3 of 6: Overloaded Methods and Static Methods / Variables

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 static variable called numEmployees that holds an int and initialize it to zero. This will allow us to count all the Employee objects created in the main class.
2. Increment numEmployees in all of the constructors
3. Add overloaded versions of setDependents and setAnnualSalary that accept strings. This way, we will have two "set" methods for both dependents and annual salary; one that accepts a string, and one that accepts its default data type.

Deliverables

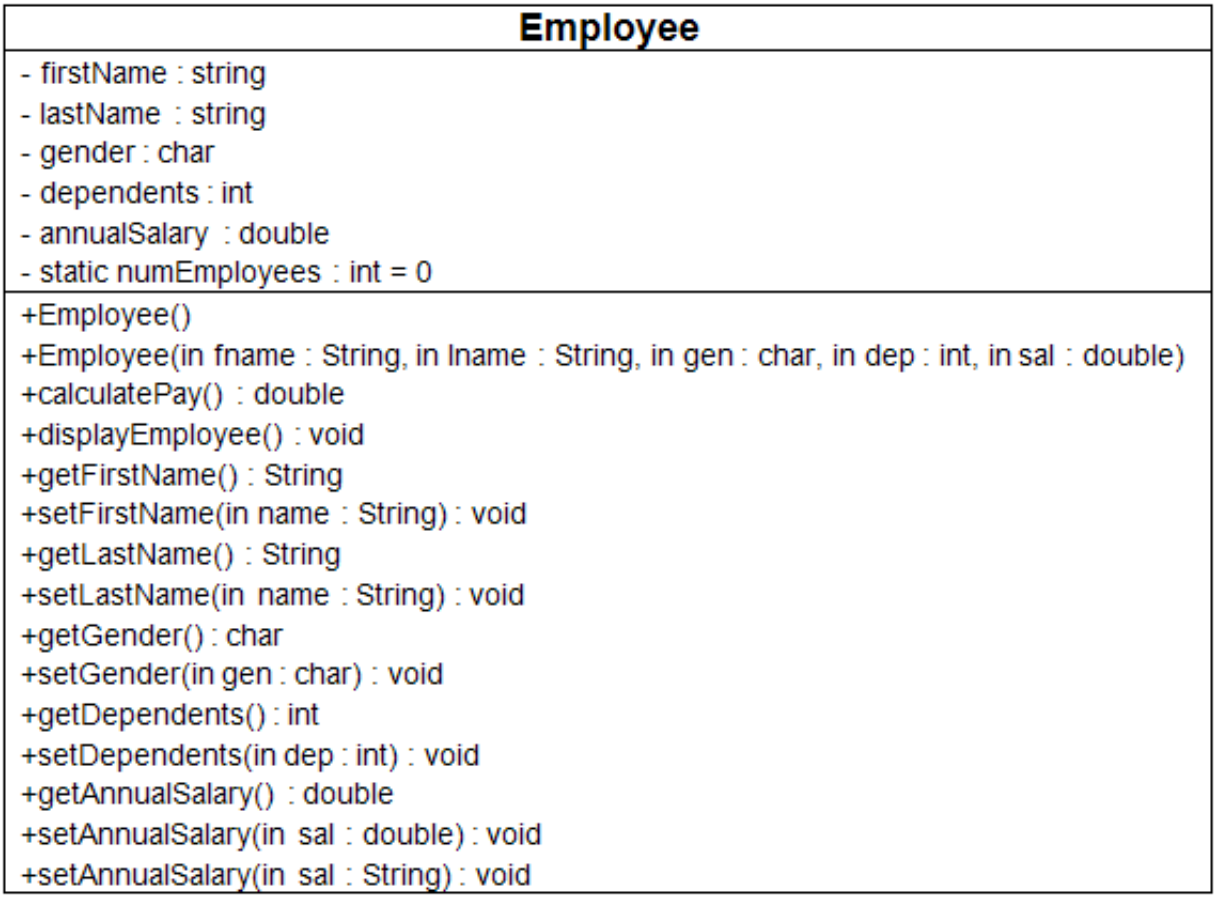
Due this week:

- Capture the Console output window and paste it into a Word document.
- Zip the project folder files.
- Put the zip file and screen shots (Word document that contains programming code and screen shots of program output) in the Dropbox.

iLAB STEPS

STEP 1: Understand the UML Diagram

Back to Top



/p>

- The following attribute has been added:

```
- static numEmployees: int = 0
```

- The following behaviors have been added:

```
+ static getNumEmployees( ) : int
+ setDependents(in dep : String) : void
+ setAnnualSalary(in sal : String) : void
```

STEP 2: Create the Project

Back to Top

You will want to use the Week 2 project as the starting point for the lab.

STEP 3: Modify the Employee

Back to Top

1. Using the UML Diagrams from Step 1, code the changes to the Employee class.
    a. Create a static numEmployees variable and initialize it to zero.
    b. Increment numEmployees by 1 in each of the constructors.

c. Create an overloaded setDependents method and this time make the parameter a string.
d. Create an overloaded setAnnualSalary method and this time make the parameter a string.
Remember that you will have to convert the string in the above two "set" methods to the data type of the attribute.
e. Make the getNumEmployees a static method. (This way, you can call it with the class name instead of an object name.)

Be sure you follow proper commenting and programming styles (indentation, line spacing, etc.).

STEP 4: Modify the Main Method

Back to Top

In the Main class, create code statements that perform the following operations. Be sure you follow proper commenting and programming styles (header, indentation, line spacing, etc.). Note that several of the steps below were accomplished in last week's assignment. New steps are in **bold**.

1. Create an Employee object using the default constructor.
2. Prompt for and then set the first name, last name, and gender. Consider using your getInput method from Week 1 to obtain data from the user for this step as well as Step 3.
3. **Prompt for and then set dependents and annual salary using the new overloaded setters.**
4. Using your code from Week 1, display a divider that contains the string "Employee Information".
5. Display the Employee Information.
6. **Display the number of employees created using getNumEmployees. Remember to access getNumEmployees using the class name, not the Employee object.**
7. Create a second Employee object using the multi-arg constructor, setting each of the attributes with the following values: **"Mary", "Noia", 'F', 5, 24000.0**
8. Using your code from Week 1, display a divider that contains the string "Employee Information".
9. Display the employee information for the second Employee object.
10. **Display the number of employees created using getNumEmployees. Remember to access getNumEmployees using the class name, not the Employee object.**

STEP 5: Compile and Test

Back to Top

When done, compile and run your code.

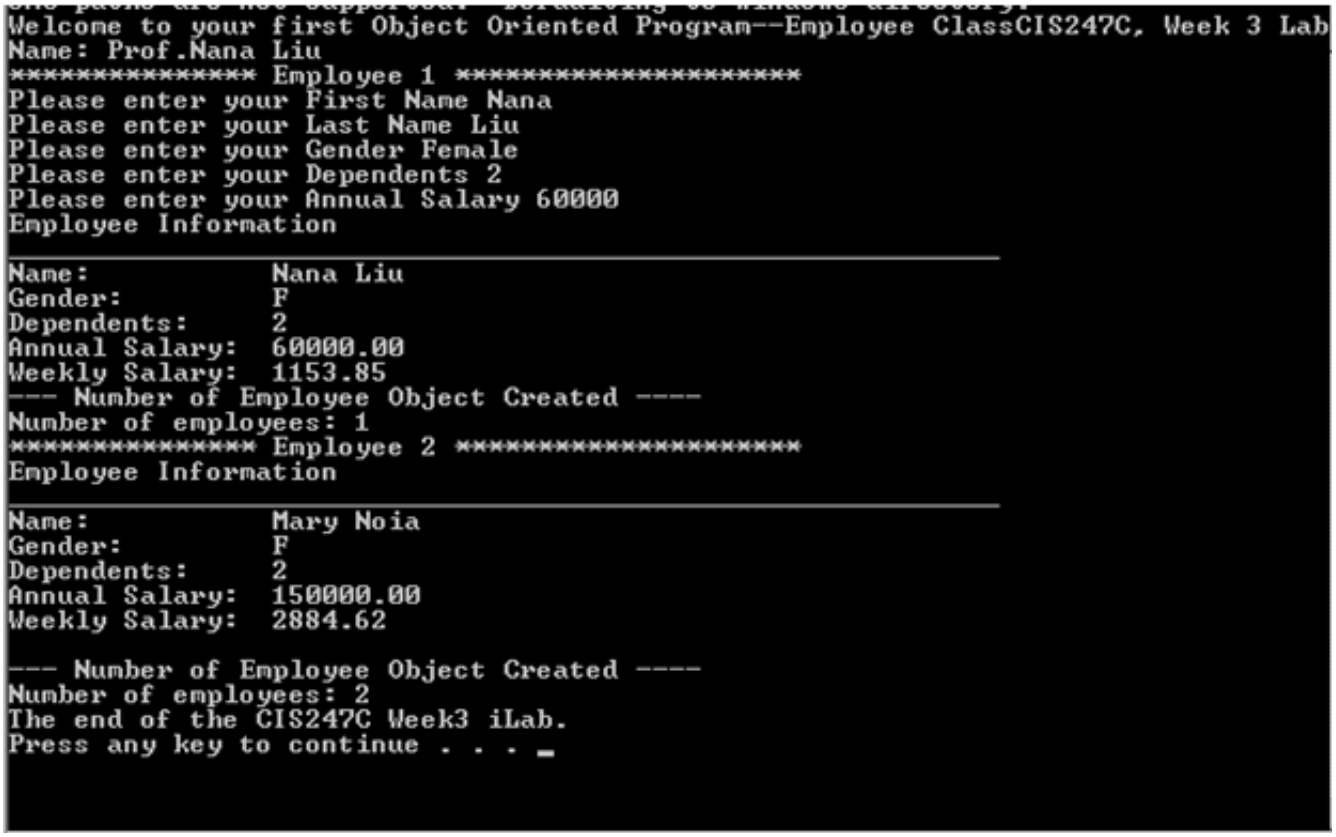
Then, debug any errors until your code is error-free.

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

STEP 6: Screen Prints

Back to Top

Capture the Console output window and paste it into a Word document. The following is a sample program output.



STEP 7: Submit Deliverables

Back to Top

- Capture the Console output window and paste it into a Word document.
- Put the zip file and screen shots (Word document that contains programming code and screen shots of program output) 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