

[Print](#)

iLab 2 of 6: Employee Class

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.](#)

i L A B O V E R V I E W

Scenario and Summary

We begin our investigation of object-oriented programming by creating an object-oriented program with a class called Employee. You will create two objects based on the Employee class, along with a class that contains the main method. The attributes, constructors, and methods for this class must satisfy the requirements in Steps 1 through 3. After you create the objects, you will prompt the user for information and then display it.

We will want to ensure that this first class is well-constructed and tested since we will extend this class in Labs 3 through 6.

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.

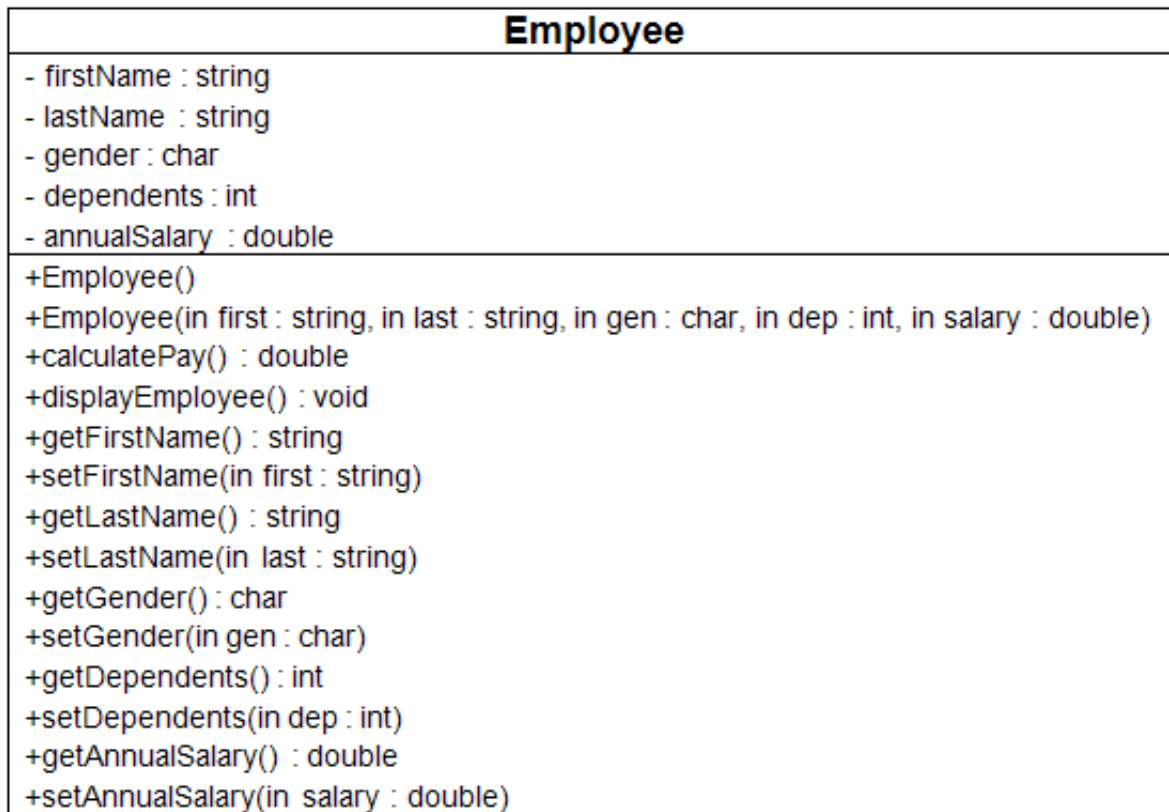
i L A B S T E P S

STEP 1: Understand the UML Class Diagram

[Back to Top](#)

Use the following UML diagram to build the class. The first section specifies the attributes. The second section specifies the behaviors, and the first character specifies the access modifier value, where:

- "-" means that the class member is private, and
- "+" means that the class member is public.



STEP 2: Code the Employee Class

[Back to Top](#)

1. Create a new project called "CIS247B_Week2Lab_YourName".
2. Using the provided Class Diagram from Step 1, code the Employee class in the new project (i.e., "Realize the UML Class diagrams").
3. The default constructor should set the attributes as follows: firstName = "not given", lastName = "not given", gender = "U" (for unknown), dependents = 0, and annualSalary = 20,000.
4. The multi-arg constructor should initialize all of the attributes using values passed in using its parameter list.
5. As shown in the Class diagram, each attribute should have a "getter" to retrieve the stored attribute value, and a "setter" that modifies the value.
6. The calculatePay() method of the Employee class should return the value of annual salary divided by 52 (return annualSalary / 52;).
7. The displayEmployee() method should display all the attributes of the Employee object in a well-formatted string with logical labels applied to each attribute. Don't forget to call calculatePay from within the displayEmployee method in order to display the Employee's weekly pay as well!

STEP 3: Code the Main Program

[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.).

1. Create an Employee object using the default constructor.
2. Prompt for and then set the first name, last name, gender, dependents, and annual salary.
(Remember that you have to convert gender, dependents, and annual salary from strings to the appropriate data type.)
3. Using your code from last week, display a divider that contains the string "Employee Information"
4. Format the currency using the following formatting services:

```
cout<<"Annual Salary:\t" <<  
setprecision(2)<<showpoint<<fixed<<annualSalary << "\n";
```

5. Display the Employee information.
6. Create a second Employee object using the multi-argument constructor, setting each of the attributes with appropriate valid values.
7. Using your code from last week, display a divider that contains the string "Employee Information".
8. Display the Employee information for the second Employee object.

STEP 4: 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 5: Screen Prints

[Back to Top](#)

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

```
Welcome to your first Object Oriented Program--Employee Class CIS247C, Week 2 Lab
Name: Prof.Nana Liu
***** Employee 1 *****
Please enter your First Name Nana
Please enter your Last Name Liu
Please enter your Gender Female
Please enter your Dependents 2
Please enter your Annual Salary 60000
Employee Information
-----
Name:      Nana Liu
Gender:    F
Dependents: 2
Annual Salary: 60000.00
Weekly Salary: 1153.85
***** Employee 2 *****
Employee Information
-----
Name:      Mary Noia
Gender:    F
Dependents: 2
Annual Salary: 150000.00
Weekly Salary: 2884.62
The end of the CIS247C Week1 iLab.
Press any key to continue . . . _
```

STEP 6: Submit Deliverables

[Back to Top](#)

- Capture the Console output window and paste 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](#)