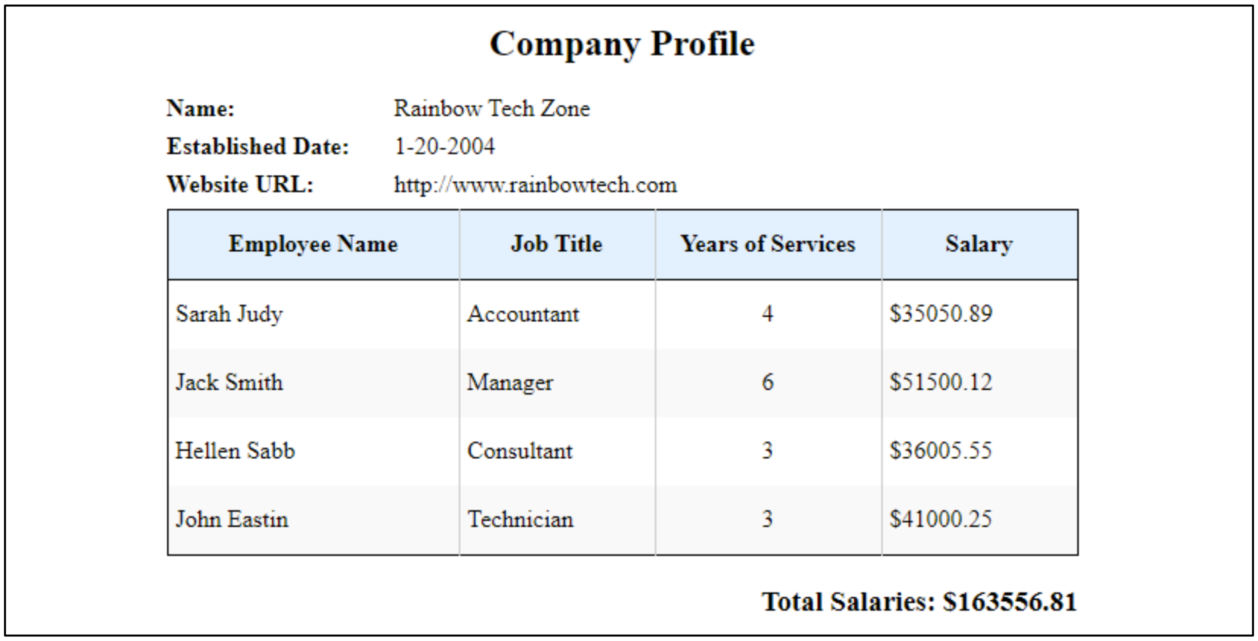
Lab 2: Creating Company Profile with OOP Objects

In this lab, you will define OOP classes to model company and employee objects. The following screenshot shows a sample output. You may try the application from the supplemental website at https://i211.sitehost.iu.edu/.



Company Profile

Name: Rainbow Tech Zone
Established Date: 1-20-2004
Website URL: http://www.rainbowtech.com

Employee Name	Job Title	Years of Services	Salary
Sarah Judy	Accountant	4	$35050.89
Jack Smith	Manager	6	$51500.12
Hellen Sabb	Consultant	3	$36005.55
John Eastin	Technician	3	$41000.25

Total Salaries: $163556.81

Two classes and three files are involved in the application. The two classes are named **Employee** and **Company**. The following are the UML diagrams of the two classes:

Employee
- name: string - title: string - years: int - salary: float
+ __construct + getName: string + getTitle: string + getYears: int +getSalary: float

Company
- name: string - established_date: string - url: string - employees: array
+ __construct + getName: string + getEstablishedDate: string + getUrl: string +getEmployee: array + getTotalSalary: float

Get ready

1. Download the data files from Canvas.
2. There is one file I've provided to you to get you started: *mycompany.php*. This is the client script that tests the **Employee** and **Company** classes. I've provided some CSS code you should use to style the web page.

Create the *employee.class.php* file

3. Create a new file and save it as *employee.class.php.* Define the **Employee** class in the file.
4. The **Employee** class models an employee. It contains four private attributes.
    a. name: the name of an employee, a string
    b. title: the job title of an employee, a string
    c. years: years of services, an integer
    d. salary: the employee salary, a float
5. The constructor should accept four values and assign them to the four attributes.
6. The four GET methods return the name, title, years, and salary, respectively.

Create the *company.class.php* file

7. Create another file and save it as *company.class.php.* Define the **Company** class in the file.
8. The **Company** class models a company. It has four private attributes.
    a. name: the company name, a string
    b. established_date: the date when the company was established, a string
    c. url: the URL of the company website, a string
    d. employees: an array of **Employee** objects. The value of the attribute is an array since a company may contain one or more employees.
9. The constructor should accept four parameters: three strings and an array of **Employee** objects.
10. The **getName, getEstablishedDate, and getUrl** methods return the company name, the established date, and the URL of the company website, respectively.
11. The **getEmployees** method returns an array of **Employee** objects.
12. The **getTotalSalary** method returns the total salary of all employees. The total salary is the sum of all employees' salaries. Since all employees are stored in an array, a loop structure is needed.

Modify the *mycompany.php* file

13. Add your code to the PHP code block for the following tasks:
    a. Require the two class files.
    b. Create an array of **Employee** objects. You may use any data. There must be four or more employees in the array.
    c. Create a **Company** object. You may use any values for company name, date, URL, and the array you created in the last step.
14. Add your code in the <body> section and below the h2 heading to display the company details. Please use the screenshot or the live demo as your guide to style your web page.

a. Use the div blocks to display the company name, date, and URL.
b. Use the table to display the company details. Make sure you use a loop to generate code for <tr> and <td> tags dynamically.
c. Use an h3 heading to display the total salary. Align the total to the right side on the page.

Code style guidelines:

1. Code style is as important as the code itself. Code style includes enough comments, adequate space between code blocks, and proper indentation, etc. Read details and view sample code from http://pear.php.net/manual/en/standards.php.
2. Please provide enough comments in your source code. Source code is a language for people, not just for computers. Comment as you go and don't wait for later. Ask yourself: "How will the next person know that?" Commenting code shows your professionalism, but also helps your grader understand your code.
3. Every class file should contain a header in this format:
   /*
   * Author: *your name*
   * Date: *today's date*
   * Name: *file name*
   * Description: short paragraph that explains what the class is for
   */
4. Indent your code. Leave enough space between code blocks.

Turning in your lab

Your work will be evaluated on completeness and correctness. Thoroughly test your code before you turn it in. It is your responsibility to ensure you turn in the correct files. You will NOT receive any credit if you turn in the wrong files.

1. Zip the entire **Lab02** folder and save it as *Lab02.zip*.
2. Upload the *Lab02.zip* file in Canvas before the lab's deadline.

Grading rubric

Your TAs will assess your lab according to the following grading rubric. You should very closely follow the instructions in this handout when working on the lab. Small deviations may be fine, but you should avoid large deviations. You will not receive credits if your deviation does not satisfy an item of the grading rubric. Whether a deviation is small or large and whether it satisfies the requirement are at your TAs' discretion. Here is the breakdown of the scoring:

Creating the **Employee** class (3 points)

Activities	**Points**

Define the four private attributes	1
Define the constructor	1
Create the four GET methods	1

Creating the **Company** class (5 points)

Activities	**Points**
Define the four private attributes	1
Define the constructor	1
Create the getName, getEstablishedDate, and getEmployees methods	1
Create the getTotalSalary method	2

Modifying the *mycompany.php* file (7 points)

Activities	**Points**
Create an array of **Employee** objects	1
Create an object from the **Company** class	1
Display the company's name, date, and URL	1
Display employee details	3
Display the total salary	1

Programming style (5 points)

Activities	**Points**
Comment your code: comments should be specific to your code	3
Use white spaces to separate code sections	1
Indent and line up code	1