University of Windsor

SCHOOL OF COMPUTER SCIENCE

COMP-2120, Fall 2024
Object Oriented Programming Using Java
LAB-5
TOTAL MARKS: 10

LAND ACKNOWLEDGEMENT

The School of Computer Science at the University of Windsor *sits on the Traditional Territory of the Three Fires Confederacy of First Nations*. We acknowledge that this is the beginning of our journey to understanding the Significance *of the history of the Peoples of the Ojibway, the Odawa, and the Pottawatomie*.

SUBMISSION DEADLINE: NOV-01, 2024

GENERAL INFORMATION

Welcome to the Lab Session of COMP-2120 where you will learn object-oriented programming (OOP) using Java. Please, arrive early to the lab and get settled to start working on the lab. Arriving 20 minutes after the start of the lab will be considered late and can affect your lab participation point. There are eight lab exercises you need to complete throughout the course. Each week on Thursday morning we will publish the lab exercise. You need to upload your code by Thursday, 10 AM in the next week.

LAB GRADING AND ASSESSMENT

You need to explain your code to TAs in the lab to receive the marks. YOU MUST demonstrate your results and explain the code to the TAs to receive marks. Thus, attending the lab sessions is mandatory. Note that your TAs are here in the Lab to help you learn. Feel free to ask questions and talk to them. Remember, the total Labs is worth 16% of your final grade.

Lab-5

The objective of this lab is to understand and apply **encapsulation** by creating an employee payroll system. The system will protect data with private fields, use getter and setter methods for controlled access, and demonstrate the concept of accessing private data only through public methods from an external class.

Part 1: Encapsulation with Employee Data (7 points)

- **Task:** Create a class `Employee` with the following private fields (attributes):
  - `String employeeName`
  - `String employeeID` (must be exactly 6 characters long, alphanumeric)
  - `double salary` (must be between 30,000 and 200,000)
  - `double bonusPercentage` (must be between 0 and 100)
  - `String address` (cannot be empty and must contain a space separating street and number)
  - `String jobTitle` (can only be "Manager", "Developer", "Designer", or "Tester")

- **Encapsulate Data:**
  - Use **private access modifiers** for all fields.
  - Provide **public getter and setter methods** for each attribute with validation:
    - `employeeName` should not be empty.
    - `employeeID` must be exactly 6 characters.
    - `salary` must be between 30,000 and 200,000.
    - `bonusPercentage` must be between 0 and 100.
    - `address` must contain at least one space to separate street name and number.
    - `jobTitle` must be one of "Manager", "Developer", "Designer", or "Tester".

- **Salary Calculation:**
  - Add a method `calculateTotalPay()` in the `Employee` class that returns the total salary, including the bonus. The formula for total pay is:
    - `totalPay = salary + (salary * bonusPercentage / 100)`

Part 2: Displaying Data via Encapsulation (3 points)

4. **Task:** Create a separate class `PayrollSystem` which will:
   - Have a static method `displayEmployeeInfo(Employee emp)` to display the employee's information.
   - This method will access the employee's data through **getter methods** of the `Employee` class (not directly accessing the fields).
5. **Task:** In the `main` method, create at least **3 `Employee` objects**:
   - One with valid values for all attributes.
   - One where invalid data is provided for `salary` or `bonusPercentage` and then corrected using the setter methods.
   - One with an invalid `employeeID` to demonstrate validation in the setter.
6. **Task:** Call `PayrollSystem.displayEmployeeInfo()` to display the employee data. This method should use **getter methods** of the `Employee` class to retrieve each piece of information.

WHAT DO YOU NEED TO DO?

1. Complete the program.
2. Attend the lab. Explain the code to TAs.
3. TAs can modify the input to your program.