**San José State University**

# Computer Science Department CS151, Object Oriented Design and Programming, 07, Spring 2020

# Homework #3

Objective:

This homework’s objective is to review and understand the unit on object oriented design principles and various ways to utilize and implement them in Java.

Details:

Exercise 1:

Design and implement a class hierarchy representing individuals involved in a business. A business might have executives, full time salaried employees, full time hourly employees, part time hourly employees, hourly paid contractors, as well as customers/clients. Each individual might have such attributes as first name, last name, age, social security number, address, level of education, payment method preference, direct deposit or not, id number (could be customer number or employee number), needs special accommodations or not, hourly pay, overtime pay, yearly salary, yearly bonus. Not all of these attributes are relevant to every type of person, while others are common to all individuals. For example, payment method preference is probably only relevant to customers. Design your class hierarchy in such a way that you only define an attribute in the appropriate parent class when possible instead of defining the same attribute in multiple child classes. Also avoid situations where child class inherits an attribute from a parent that should not belong to the child. In other words, children should only inherit from appropriate parent classes. If you need to, introduce multiple levels of inheritance. The address should not be a simple *String* object. It should be a separate class with individual fields defined for street number, street name, city name, zip code, and state. For this exercise, we will assume that no person has an apartment number or a second address line.

Each class needs to have the following constructors: default, accepts first and last name as input arguments, accepts first and last name as well as address as input arguments. Each class also needs to have getters and setters defined and implemented for all the attributes.

Implement *toString()* method in each of the leaf classes: executive, full time salaried employee, full time hourly employee, part time hourly employee, hourly paid contractor, and customer. This method should return a *String* object that contains text information about all the attribute values, including the type of person, for that class instance. Also implement method *introduce()* that displays the information returned by *toString()* method to command line. Implement a variation of method *introduce()* to flag if social security number should be displayed or not. Introduce another variation of this method to flag if both social security and pay information should be displayed.

Implement *computePay()* method in each of the leaf classes, except customer class. For an executive, this method will not accept any input arguments. For a full time salaried employee, this method will accept the number of weeks as an input argument (this input can be either an integer value or a floating point number). For a part time hourly employee, a full time hourly employee, or an hourly paid contractor, this method will accept the number of hours as an input argument. For all of these individuals, this method will return a floating point value indicating the value of the pay for the specified input arguments. Remember that the logic of computing the pay is different for the different individuals. For an executive, the pay is simply their yearly compensation plus the yearly bonus. For a full time salaried employee, the pay is based on their yearly compensation adjusted to the number of weeks specified by the input argument. For a full time hourly employee, the pay is based on the hourly pay and the number of hours specified by the input argument. Make sure to account for possible overtime. For a part time hourly employee, the pay is based on the hourly pay and the number of hours specified by the input argument. Part time hourly employees are not allowed to work more than 40 hours a week, so implement a check for the proper number of hours and if they exceed 40 hours, return -1. For an hourly paid contractor, the pay calculation is the same as for full time hourly employees.

Implement *makePayment()* method only available to customers. This method should display the preferred method of payment for the customer to command line screen.

The instructions for class hierarchy are deliberately vague, allowing you some freedom in how you design your application. The goal is to make you put in some thought into how to best design your classes and their relationships. The objective here is to practice object oriented design principles: abstraction, inheritance, encapsulation, and polymorphism. Do not forget to ask yourself questions such as “which one of these is a person?” or “which one of these is an employee?” or “which ones of these are business associates and which ones are not?” Think your design through. Maybe draw the relationships on a piece of paper before implementing to help yourself visualize the hierarchy of classes.

Define and implement class **BusinessTest**.This class should implement *main()* method. In the body of the *main()* method you should create at least 2 instances of each of the leaf classes: executive, full time salaried employee, full time hourly employee, part time hourly employee, hourly paid contractor, and customer. It is up to you to choose which values of the attributes for each of the instances to use. For each instance, make a call to *introduce()* method and make a call to *computePay()* method for business associates and *makePayment()* for customers. Again, it is up to you to choose the input arguments into *computePay()* method where they are required. For visual presentation make sure to include an empty line between each employee instance output. Save this class and its definition into a file named **BusinessTest.java**.

It is strongly suggested that in addition to **BusinessTest.java** file you submit the following files for this homework assignment: **Person.java**, **Customer.java**, **Employee.java**, **Executive.java**, **FullTimeSalaryEmployee.java**, **FullTimeHourlyEmployee.java**, **PartTimeHourlyEmployee.java**, **Contractor.java**.

Submission:

In your class repo create a directory called “Assignment3” and add all the files created for this homework assignment to that directory.

This homework assignment is assigned on 02/11/2020 and is due on 02/20/2020 before 11:59pm. Email your assignment submission to me at both [Yulia.Newton@sjsu.edu](mailto:Yulia.Newton@sjsu.edu) and [yulia.newton@gmail.com](mailto:yulia.newton@gmail.com), as well as the grader at [akshay.kajale@sjsu.edu](mailto:akshay.kajale@sjsu.edu). The subject of the email should say “CS151 Assignment 3”. Add your name as it appears on the class roster and the URL to your git repo in the body of an email.

Grading:

Your code must compile and execute successfully in order to get full credit for this assignment. I will compile and execute BusinessTest.java file listed in that exercise description.

* Program with no compile errors (10 pts)
* Program executes (10 pts)
* Program outputs what is required by the exercise (10 pts)

A total of 30 points are possible for this homework assignment.