2021F CS234 Computer Science II

Lab 7
Total points: 100

E9.12

For this laboratory you need to create the definition of **three** Java **classes.** You will practice:

- Understanding basic UML class diagrams (you will need this for your project)
- inheritance (i.e., extend, super, overriding).

For this lab you need to **implement** the following **classes** (no main methods PLEASE!!)

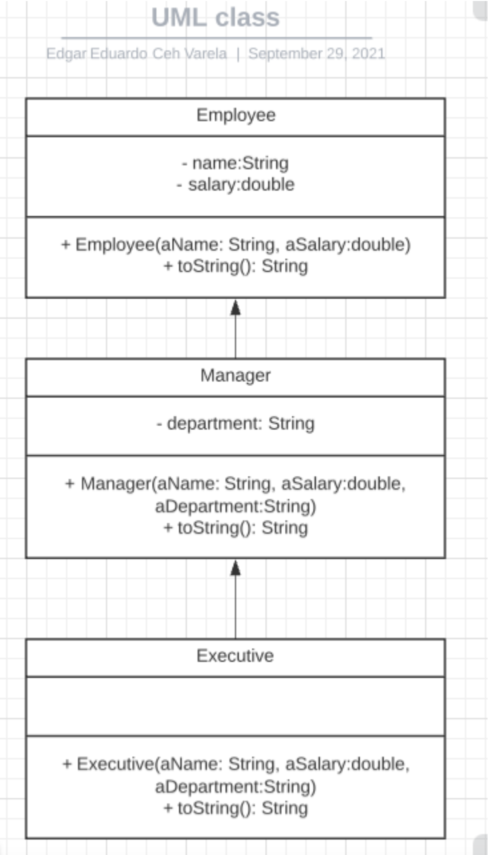

UML class
Edgar Eduardo Ceh Varela | September 29, 2021
Employee
- name:String
- salary:double
+ Employee(aName: String, aSalary:double)
+ toString(): String
Manager
- department: String
+ Manager(aName: String, aSalary:double, aDepartment:String)
+ toString(): String
Executive
+ Executive(aName: String, aSalary:double, aDepartment:String)
+ toString(): String

Diagram elaborated in Lucidchart. It is free! www.lucidchart.com

About the **toString()** method.

It is a **built-in method** in Java that **returns** the **value** given to it in **string** format.
Hence, **any object** that this method is applied on, will then **be returned as a string** object.

In other words, it is a method that can be used to **return** anything as a String.
Because it is built-in in the default Object class (all classes by default inherit this class) you need to override it.

More about toString()
https://www.educative.io/edpresso/how-to-use-the-tostring-in-java

Therefore, your **toString** method needs to **return** a **string** with the **information** necessary to be **used** with a **print** statement.

How to **test** your program?
You can implement a **tester program** like the following:

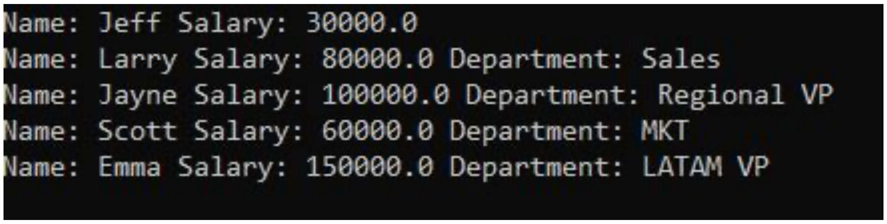
```
public static void main(String[] args)
   {
      Employee e1 = new Employee("Jeff", 30000);
      Employee e2 = new Manager("Larry", 80000, "Sales");
      Employee e3 = new Executive("Jayne", 100000, "Regional VP");

      Manager m1 = new Manager("Scott", 60000, "MKT");
      Executive ex1 = new Executive("Emma", 150000, "LATAM VP");

      System.out.println(e1);
      System.out.println(e2);
      System.out.println(e3);
      System.out.println(m1);
      System.out.println(ex1);

   }
```

The **output** of the **tester** program is the following:



```
Name: Jeff Salary: 30000.0
Name: Larry Salary: 80000.0 Department: Sales
Name: Jayne Salary: 100000.0 Department: Regional VP
Name: Scott Salary: 60000.0 Department: MKT
Name: Emma Salary: 150000.0 Department: LATAM VP
```

Please, **after** working in your program, **answer** the following question.
Why can we use the class Employee or the class Manager when creating a new Manager?
Write your **answer** in the **PDF** with your screenshots.

Submission details:

Upload a **single ZIP** file.
Name your file as follows: **Lab7_Lastname_Firstname.zip**
There is a **10% points deduction** if your file does not have the correct name.

Your **.zip** file must contain the following:

1. Your **.java** source files for your **classes** (.java files **without** the **main** method. No .class files).
2. A **SINGLE PDF** with **screenshots** from your programs running **AND** your **answer** to my **question** (10% points deduction if you don't submit a **SINGLE** PDF file)

For this lab, you do **not need to submit the .txt** file with your instructions. **Why?** Because I will use my tester program to use your classes. Therefore, it is **extremely important** that your **class and method names** are **correct.**

I don't require the use of packages. **However**, **If**, you want to use a package to organize your code then you **MUST submit a .txt** file with **clear instructions** about **how to use** your package (i.e., the package name, the package path, imports, how to compile, and how to execute your program, etc.). Review the lecture from 09/23/2021 about packages.

In each .java file, **write** as a multiline comment at the beginning of the file the following:

1. Your name
2. The ID of the problem (e.g., P8.15)
3. The course section

The **zip** file must be uploaded to Canvas. I do not accept answers via email.
I do not accept image files; it must be a **PDF** file.

Make sure to check the **due date** for this activity on Canvas.
Make sure you are **submitting the correct files**. I will grade the file uploaded to Canvas.
Make sure you **test** your **classes** with a similar **tester program** as the one I am showing in this lab (i.e., a .java program with a main method where you create objects from your class). Please, **test your program** with different use cases.
Use the ***javac*** and ***java*** commands to compile and **test** your classes **before** submitting your solution.
You **don't submit** your **tester** program. No classes with main methods.
Make sure to review the grading rubric.
Late submissions are not allowed.