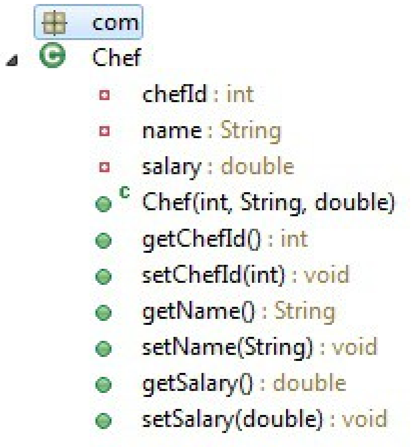
PS1 Diagnostics

Note:

- Go through the problem statement carefully.
- Time limit is 90 minutes.
- Make sure that project is created in eclipse only.
- Create all your java files in package **“com” within src folder** of eclipse project.
- Make sure that exact class outline is followed as you did in previous assignments.
- You need to zip the eclipse project folder and upload the same in LMS once completed. The project folder will be available in your workspace folder.
- It is mandatory to upload eclipse project and not java files for your code to be assessed.
- Make sure that there is no compilation error in your code before submission. Even if there is minor error, entire solution could be rejected.
- You may refer previous assignments, course content and internet for any reference.
- Create a separate class(with any name) and write the main method in it so that you can test your code.

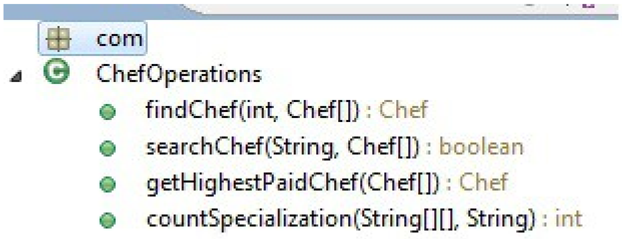
Create an application to manage the details of Chef in a hotel. The features that have to be included in the application are explained in the later sections.

Create a Chef class as shown in the below outline.



```
com
Chef
    chefId : int
    name : String
    salary : double
    Chef(int, String, double)
    getChefId() : int
    setChefId(int) : void
    getName() : String
    setName(String) : void
    getSalary() : double
    setSalary(double) : void
```

Create a class ChefOperations as shown in the below outline. The details of the functions that have to be implemented in the class are explained after the outline.



com
ChefOperations
findChef(int, Chef[]) : Chef
searchChef(String, Chef[]) : boolean
getHighestPaidChef(Chef[]) : Chef
countSpecialization(String[][], String) : int

findChef: This method takes the following inputs, the chefId of a Chef and an array of Chef objects. If a Chef with the given chefId is found from the array of Chefs, then this method returns that found Chef object else returns null.

searchChef: This method takes the name of a Chef and an array of Chef objects as input. This method has to check if any Chef with the given name is present in the array of Chef objects. If found then return true else return false. Please note that the search has to be case-insensitive.

getHighestPaidChef: This method takes an array of Chef objects as input and returns the details of the Chef who is having highest salary.

countSpecialization(OPTIONAL): This method has to count the number of specializations for a given Chef. This method takes following inputs, 1. a two dimensional array of strings representing the name of the Chef and his/her specialization and 2. name of the Chef for whom the number of specializations have to be counted. Please see the example for details.

Index	0	1	2	3
0	John	Italian		
1	Samantha	Italian	Continental	
2	Jack	Chinese		
3	Philip	Italian	Chinese	Continental

The zeroth index on each row represents the name of the Chef and from 1st index onwards represent the kind of specialization he/she is having. Hence John has 1 specialization, Samantha has 2 specializations , Jack 1 and Philip 3 specializations respectively.

So if Philip is given as input along with the above array of specializations, then the output from the method has to be 3.