

Write Java code as instructed.

- Define an interface **Appraisable** that has the following members:
 - Default method `default void appraisal(Teacher t)` that increments the salary of an **Employee** by $(stuPassPer/100)*5000$.
 - Abstract method `public abstract void checkAndUpdateSalary()`
- Define an interface **SpecialAppraisable** that extends **Appraisable** and has the following members:
 - Default method `default void spAppraisal(Teacher t)` that increments the salary of an **Employee** by $(stuPassPer/100)*10000$.
- Class **Teacher** that implements the interface **SpecialAppraisable** and has the following members:
 - **String** `name`, **double** `salary` and **private double** `stuPassPer` as private instance variables
 - Constructor to initialize the instance variables
 - Mutator method to update `salary`
 - Accessor method to access `salary`
 - Accessor method to access `stuPassPer`
 - Override method `toString()` that returns `name`, `salary` and `stuPassPer` of the **Teacher** as a single concatenated string (each separated by a single space)
 - Overriding method `public void checkAndUpdateSalary()` that has the following functionality.
 - * If `stuPassPer >= 60` and `stuPassPer < 75` then invoke the `appraisal()` method from **Appraisable** interface
 - * Else, if `stuPassPer >= 75` and `stuPassPer <= 100` then invoke the `spAppraisal()` method from **SpecialAppraisable** interface

- Class `InterfaceTest` that has the following members:
 - You should define method `public static void printUpdatedTeachList(Teacher[] tList)` that has the following functionality
 - * Iterate over array `tList` and invoke method `checkAndUpdateSalary()` on each `Teacher` object.
 - * Then, iterate over `tList` and display each `Teacher` object.
 - `main` method has the following functionality
 - * Creates and initializes an array `tArr` of three `Teacher` objects
 - * Invokes method `printUpdatedTeachList(Teacher[] tList)` to print the updated details of each `Teacher` after the appraisal is applied

Test Run Results

[Down](#)

Test Case 1

Input	Expected Output	Actual Output
Akshay 50000 64.5	Akshay, 53225.0, 64.5	
Anasuya 60000 80.00	Anasuya, 68000.0, 80.0	
Abhaya 75000 70	Abhaya, 78500.0, 70.0	

Test Case 2

Input	Expected Output	Actual Output
Dharani 76000 96	Dharani, 85600.0, 96.0	
Thanvi 56000 72	Thanvi, 59600.0, 72.0	
Thannya 90000 85	Thannya, 98500.0, 85.0	

Submission Results

Test Case 1

Input	Expected Output	Actual Output
Kavya 54000 59	Kavya, 54000.0, 59.0	
Chaithanya 65000 55	Chaithanya, 65000.0, 55.0	
Sravani 73000 53	Sravani, 73000.0, 53.0	