|  |  |
| --- | --- |
| **CSE1206 : Object Oriented Programming Lab Fall 2018** | **Set- B** |

**Online: 2 Date: 29 January, 2019 Group: B1 Time: 35 minutes**

**ID:**

**Marks**

|  |  |
| --- | --- |
| 1. Create a Java Project Named **‘EmployeeSalary’**. Inside this project (folder) create a new class named **‘Salary’**. (There should be two classes: **EmployeeSalary** [the default class], **Salary** [the newly created class]. And the two classes should be in the same package). | **1** |
| 1. Inside the **Item** class declare **3** **private** variables: **position (String), totalSalary (double), raise (double). [raise is the percentage of salary that has increased]** | **1** |
| 1. Declare **2 public** Constructorsin **Item Class:**     1. A consturctor that takes no parameter and initializes the variables as:   **position** = null  **totalSalary**= -1.0  **raise** = -1.0   * 1. Another Constructor which takes all the variables as parameter and assigns those parameters to the class variables. | **2+2** |
| 1. Declare necessary **public** getter() and setter() methods for all the 3 variables. Set the **totalSalary** using its **setter method** in such a way that it can never be less than or equal to zero. If zero or less than zero is assigned then set it to 1. | **1+2** |
| 1. Declare a **public** method called **changeSalary()** in **Salary Class** that returns a **double** value and takes no parameters. The method calculates the final salary after applying the percentage raise. **For example**: if totalSalary is 1000 and percentage **raise** is 20, then 20% of 1000 is 200. So total salary after applying raise is 1200. | **3** |
| 1. Declare another **public** method called **compareSalary()** in **Salary Class** which takes the **Salary object** as parameter and returns a **String** variable. This method compares the final changed Salary of two separate employees after calculating the change and returns the following Strings:  * If both salaries are same then return “equal” * If salary of employee-1 is more than employee-2; return “greater” * If salary of employee-1 is less than employee-2; return “smaller”   (employee-1= Salary Object of Current Class , employee-2= Salary Object of the Class in the Parameter) | **3** |
| 1. Now test the methods of **Salary** Class in the **EmployeeSalary** Class. Take user input for employee **postion**, **totalSalary** and **raise**. Check the final salaries obtained after applying **raise** and print those with the particular postions.   Then set the **raise** of the employee-1 to **68** using the setter method and again compare the salaries of employee-1 and employee-2. | **5** |

**Total: 20**

|  |  |
| --- | --- |
| Sample Input 1 | Sample Output 1 |
| Enter Position of Employee-1: Manager  Enter Salary of Employee-1: 45000  Enter Raise for Employee-1: 30  Enter Position of Employee-2: HR  Enter Salary of Employee-2: 60000  Enter Raise for Employee-2: 25 | Changed Salary of Manager is 58500.0  Changed Salary of HR is 75000.0  Salary of Manager is less than HR  After Changing the Percentage for Manager:  New Changed Salary of Manager is 75600.0  Salary of Manager is greater than HR |

**Hints:**

\*user Scanner class to take input:

for string use nextLine() , for double use nextDouble()

\*When you wrtie String a= sn.nextLine()

String b=sn.nextLine()

After taking the first input when you press enter the **enter (“\n”)** that is the newline will be taken as the input for String b which you don’t want. To solve this problem you can insert another sn.nextLine() between them like this:

String a= sn.nextLine()

Sn.nextLine()

String b=sn.nextLine()

So after you take the first input and press enter the middle nextLine() will take the newline input and then won’t effect your String b input. Do this after each String if you have more inputs.