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

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

**ID:**

**Marks**

|  |  |
| --- | --- |
| 1. Create a Java Project Named **‘TestTaxi’**. Inside this project (folder) create a new class named **‘Taxi’**. (There should be two classes: **TestTaxi** [the default class], **Taxi** [the newly created class]. And the two classes should be in the same package). | **1** |
| 1. Inside the **Item** class declare **3** **private** variables: **destination (String), distance (double), fairPerKm (double).** | **1** |
| 1. Declare **2 public** Constructorsin **Item Class:**     1. A consturctor that takes no parameter and initializes the variables as:   **destination** = null  **distance** = -1.0  **fairPerKm** = -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 **distance** 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 **computeTotalFair()** in **Taxi Class** that returns a **double** value and takes no parameters. The method calculates the total fair for the total distance. **But it also gives 20% discount to all riders.**   **For example**: if distance is 7 km and fairPerKm is 50tk , then total fair is 350tk.  20% discount on total fair of 350tk is 70tk. So final fair is 280tk. So return 280tk**.** | **3** |
| 1. Declare another **public** method called **compareFair()** in **Taxi Class** which takes the **Taxi object** as parameter and returns a **String** variable. This method compares the total fair of two separate taxi rides and returns the following Strings:  * If both fairs are same then return **“same”** * If fair of taxi-1 is greater than taxi-2; return **“more”** * If fair of taxi-1 is lesser than taxi-2; return **“less”**   (**taxi-1**= Taxi Object of Current Class , **taxi-2**= Taxi Object of the Class in the Parameter) | **3** |
| 1. Now test the methods of **Taxi** Class in the **TestTaxi** Class. Take user input for  **destination**, **distance** and **fairPerKm**. Check the total fair for a particular ride and print those with their respective destination.   Then set the **fairPerKm** of the taxi-1 to **20** using the setter method and again compare the fair of taxi-1 and taxi-2. | **5** |

**Total: 20**

|  |  |
| --- | --- |
| Sample Input 1 | Sample Output 1 |
| Enter Destination of Taxi-1: Uttara  Enter Distance of Taxi-1: 10  Enter Fair per kmn of Taxi-1: 30  Enter Destination of Taxi-2: Mirpur  Enter Destination of Taxi-2: 5  Enter Destination of Taxi-2: 45 | Total fair to Uttara is 240.0  Total fair to Mirpur is 180.0  Total fair to Uttara is more than Mirpur  After Changing the fair per kmn to Uttara:  New Total fair to Uttara is 160.0  Total fair to Uttara is less than Mirpur |

**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.