Sapient Hiring Exercise – Interviewer Guide

_____ _____

ne: _____ _____

ne: _____

Instructions

- Duration of this exercise is 150 minutes. Please manage your time accordingly. There isn't any stipulated duration; only a guidance. However, sooner you complete the exercise, more marks you will get. Please work accordingly
- Make any necessary assumption, and clearly state the assumptions made.
- Do not seek any help – online or through any other source.

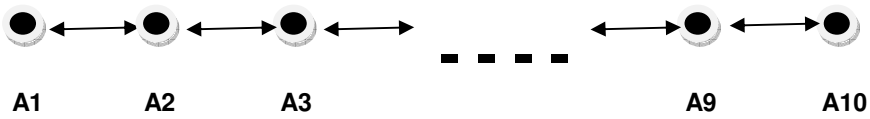
Expectations

- Write the compiling code using your choice of IDE.
- Wherever necessary, provide Unit Test Cases using JUNIT (if you're not conversant with JUNIT, just list down unit test cases)

Problem Statement

Sapient has been asked to implement 'Metro Smart Card System' (MSCS) for Delhi city. For this application assume there is a single metro line covering 10 stations linearly.

The stations name are A1, A2, A3, A4, A5, A6, A7, A8, A9, A10 as shown below. The travel can be in any direction.



Travelers have smart cards that behave just like any regular debit card that has an initial balance when purchased. Travelers swipe-in when they enter a metro station and swipe-out when they exit. The card balance is automatically updated at swipe-out.

Objective

Objective of the exercise is to create an automated system that has following functionalities:

Card should have a minimum balance of Rs 5.5 at swipe-in. At swipe-out, system should calculate the fare based on below strategies set at the start of the day. The fare must be deducted from the card. Card should have the sufficient balance otherwise user should NOT be able to exit.

Weekday – Rs. 7 * (Number of stations traveled)
Weekend – Rs. 5.5 * (Number of station traveled if it's Saturday or Sunday)
(* there can be more such fare strategies in future)

Additionally, system needs to have functionality to generate some statistics/reports defined below. So, system needs to provide following APIs

- API to get total foot-fall (swipe-in + swipe-out) for a given station.
- API to generate a "per-card report" on demand: It should prints the following information on console:
  Card <number> used to travel from station <source_station> to station <destination_station>. Fare is Rs <x> and balance on the card is <Rs x>

Evaluation criteria

- Code Completeness/ Correctness
- Code Structure and quality: Modularity, usage of OO principles, size of classes/functions, class/function/variable names, package/class structure
- Choice of data structures
- Unit Test cases
- Coding productivity (more time you take to submit the exercise, lesser you will score)

Better you perform on this criteria, higher you will score.