**PRO192- Practice Exam – Fall 2023 05/11/2023**

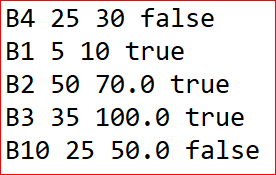
Duration : 85 minutes

Develop Bi\_Store Management OOP application with following components and requirements

1. Define class **Bicycle** in package model with following attributes and methods: **(2 marks)**
   1. Attributes: code , speed , distance , is electric bicycle . (0.5 marks)
   2. Constructors, getter /setter (0.5 marks)
   3. Define method **needToRecharge** which returns a string "Need to recharge batteries" in case the bicycle is electric and the distance greater than 60 km, otherwise, it will return the string "Don't need to recharge". (0.5 marks)
   4. Override method **toString** which return a string follow the giving format: (0.5 marks)  
      For electric bicycles, add prefix E- before class name Bicycle,

**Ex**: E-Bicycle |B1 |5 |10.0 |Don’t need to recharge

1. Define class **Bi\_Store** in package model to manages the list of bicycle with following methods: **(4 marks)**
   1. Read text file to load data file InputFile.txt as following informations: (1.5 mark)



* 1. Add new bicycle into the list (0.5 marks)
  2. Delete an existed bicycle in the list by its code (0.5 marks)
  3. Search for bicycle which takes the least time to move. timetomove=distance/speed 0.5

1. Define **StoreManagement** class in package controller which following requirement: **(4 marks)**
   1. Display a Menu of the following choices: Display all bicycles, Add new bicycle, Delete a bicycle, Searching for the least time to move, Quit application. (1.5 marks)   
      ***Hint****:* An abstract Menu class is provided for use in this application.
   2. Method doAddBicycle() gets bicycle properties from keyboard to add new bicycle into Bi\_Store class:

* Validate for **code** starts with "B" and follow by digit characters (Example: B13) with no duplicate code in the list (1 marks)
  1. Method doDeleteBicycle() that gets bicycle code from keyboard and delete that bicycle from the list in Bi\_Store class.(0.5 marks)
  2. Method displayAll() display list of bicycle sorted ascendent by speed.(0.5 marks)
  3. Main function to run the application. (0.5 marks)

Notice: Students must strictly follow all requirements as described above for marking. Any modification of the requirement details is not allowed and will receive no marks!

**User interface of first choice as sample images below:**

