**CSD201 PE INSTRUCTIONS**

**Follow the steps below to complete PE:**

1. Create a folder to save given projects, e.g. CSD\_given (1). Download the given materials to (1).
2. Complete the requirements of the test.
3. **Before submission:**
   1. **Clean and Build Project** (Shift+F11),
   2. Then rename the folder **dist** to **run**. (If the folder run already exists, delete it before renaming).
4. **Submission:**
   1. To submit the project Q1, at first you must select the Question No: 1
   2. Browse and select the project folder (e.g. Q1, or Q1A, or Q2, …)
   3. Then click the **Submit** button.
5. **Do not use accented Vietnamese** when writing comments in programs.
6. **Do not add** new **import** statement(s) to given files.
7. Software tools must be used: **NetBeans IDE 13** and **Java JDK 1.8**.

**If at least one of the above requirements is not followed, the exam will get ZERO.**

* ***Notes:***
  + ***The input and expected output below are only used to test your codes.***
  + ***The input and expected output in the real testcases (for marking) are different with in the examples below.***
  + ***Do not hardcode with the given expected results.***

**Troubleshooting:**

If the given project (e.g. Q1) runs with error, you need to run "**Clean and Build Project**" (Shift+F11). If still error, try to rename or copy the project to other one, e.g. from Q1 to Q1X or Q1Y.

**Questions: (10 marks)**

The given files already contain statements to implement a program for managing a computer store. The structure of the main classes is as follows:

* Class **Car**: Contains information about a Car object, including CarID, color, and price.
* Class **Node**: Includes a Car object and a next pointer for linking. Used by CarQueue.
* Class **TreeNode**: Includes a Car object and left, right pointers for the tree structure. Used by CarBST.
* Class **CarQueue**: Is a queue structure (implemented as a linked list of Nodes). This Queue is used to manage incoming Cars following FIFO (First-In, First-Out) method.
* Class **CarBST**: Is a Binary Search Tree structure, where the TreeNode contains complete Car data. This Tree's role is to store, search, and sort Car data alphabetically by CarID (in-order traversal).
* Class **CarStore**: Is the main class of the program, containing a CarQueue and a CarTree.

Students are required to carefully read the provided code segments to fully understand the relationships between the classes and the functions within each class. The specific task of the test is to execute the following requirements:

1. **f1() – 3 marks** : To complete the requirement f1, students need to fulfill these two specific tasks: implement the method **enqueue()** in **CarQueue** and the method **insert()** in **CarBST**.
   * + enqueue() follows the principle of Queues.
     + insert() follows these following rules:
       - Cars are inserted based on their CarID (alphabetically comparison)
       - If a Car with same ID exists, overwrite it with new Car.

*The expected output used to test your code are as follows:*

Car Queue: (C001,green,10925.0) (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C009,silver,1498.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0) (C009,silver,1498.0)

1. **f2() – 2 marks**: To complete the requirement f2, students need to perform this specific task: implement the **search()** method of **CarBST** with the given **searchID**, then:
   * + Print the ***returned Car object*** if found.
     + Or ***null*** if not found.

***Example:***

***Case of found ID:***

Searching for Car ID in BST: C009

*The expected output used to test your code are as follows:*

Car Queue: (C001,green,10925.0) (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C009,silver,1498.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0) (C009,silver,1498.0)

Found: (C009,silver,1498.0)

***Case of ID not found:***

Searching for Task ID in BST: NoID

*The expected output used to test your code are as follows:*

Car Queue: (C001,green,10925.0) (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C009,silver,1498.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0) (C009,silver,1498.0)

Car not found in BST

1. **f3() – 2 marks**: In this task, students need to perform the **findMax()** method in **CarBST**. This function will printout the Car with the highest “alphabetical” **CarID**.

*The expected output used to test your code are as follows:*

Car Queue: (C001,green,10925.0) (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C009,silver,1498.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0) (C009,silver,1498.0)

Highest Alphabetical Car ID in BST: (C009,silver,1498.0)

1. **f4() – 3 marks**: In this task, students need to be able to perform these following methods:
   * + **dequeue()** in **CarQueue** method is used to dequeue the front Car in **CarQueue**.
     + **remove()** in **CarQueue** method will remove the Car in **CarQueue** with given **CarID.**
     + **remove()** in **CarBST** method will remove the Car in **CarTree** with given **CarID**.

***Example:***

Dequeued from Car Queue: C001 (Implement)

Delete C009 from both Queue and Tree:

*The expected output used to test your code are as follows:*

Car Queue: (C001,green,10925.0) (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C009,silver,1498.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0) (C009,silver,1498.0)

Car Queue: (C006,black,28000.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C007,red,19999.0)

Car BST: (C001,green,10925.0) (C003,silver,99000.09) (C004,pink,31111.99) (C005,yellow,2600.0) (C006,black,28000.0) (C007,red,19999.0)