

CSD201 PE INSTRUCTIONS

Read the instructions below carefully before start coding.

Students are ONLY allowed to use:

- Materials on his/her computer (including JDK, NetBeans...).
- For distance learning: Google Meet, Hangout (for Exam Monitoring Purpose).

Follow the steps below to complete PE:

1. Create a folder to save given projects, e.g. CSD_given (1). Down load given materials to (1).
2. Steps to do question 1 (do the same for questions 2 and 3): Open NetBeans, open the given Q1 project, then edit the MyList.java file according to the requirements of the exam. (edit the file BSTree.java for Q2 and Graph.java for Q3).
3. Before submission: Run the function "**Clean and Build Project**" (Shift+F11), to ensure BUILD SUCCESSFUL (if not, the project will get 0 mark).
4. **Submission:** to submit the project Q1, at first you must select Question No = 1, browse and select the project folder (e.g. 1, Q1 or Q1X,...) then click the **Submit** button. Do the same for other questions. **Do not submit** the un-edited given project. If project is too big for submission, delete all f1.txt, f2.txt,....
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 8.x** and **Java JDK 1.8**.

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

Trouble shooting:

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,...

Question 1: (4 marks)

(Do not pay attention to real meaning of objects, variables and their values in the questions below).

In this question you should complete some methods in **MyList.java** file.

The class Car with 2 data members: owner and price is given and you do not need to edit it. The MyList class is a linked list of Car objects. The following methods should be completed:

- void addLast(String xOwner, int xPrice) - check if xOwner.charAt(0) = 'B' or xPrice>100 then **do nothing**, otherwise add new car with owner=xOwner, price=xPrice to the end of the list. (price can get arbitrary value, even negative).
- void **f1()** – This method is used to test the addLast methode above. You do not need to edit this function. Output in the file **f1.txt** must be the following:

(A,9) (C,7) (D,2) (E,6) (F,4)

- void **f2()** – There is a given objects x. You should write statements so that x will be the first element of the list. Output in the file **f2.txt** must be the following:

(C,9) (D,6) (E,8) (F,2) (I,6)

(X,1) (C,9) (D,6) (E,8) (F,2) (I,6)

- void **f3()** – Suppose the list contains at least 3 elements. Delete the first node having price=5. Output in the file **f3.txt** must be the following:

(C,9) (**D,5**) (E,3) (F,5) (I,6)

(C,9) (E,3) (F,5) (I,6)

- void **f4()** – Sort the list ascendingly by price. Output in the file **f4.txt** must be the following:

(C,9) (D,6) (E,5) (F,13) (I,2) (J,1)

(J,1) (I,2) (E,5) (D,6) (C,9) (F,13)

Question 2: (4 marks)

In this question you should complete some methods in **BSTree.java** files.

The class Car with 2 data members: owner and price is given and you do not need to edit it. The BSTree class is a binary search tree of Car objects. The variable **price is the key of the tree**. The following methods should be completed:

- void insert(string xOwner, int xPrice) - check if xOwner.charAt(0) = 'B' or xPrice>100 then **do nothing**, otherwise insert new car with owner=xOwner, price=xPrice to the tree.
- void f1() – You do not need to edit this function. Your task is to complete the insert(...) function above only. Output in the file **f1.txt** must be the following:

(A,5) (C,2) (E,4) (G,3) (D,6) (F,7)
(C,2) (G,3) (E,4) (A,5) (D,6) (F,7)

- void f2() – Perform pre-order traversal from the root but display to file f2.txt nodes having price in the interval [3,5] only. **Hint:** Copy the function preOrder(...) to preOrder2(...) and modify it. Output in the file **f2.txt** must be the following:

(C,6) (D,2) (F,4) (H,3) (I,5) (E,8) (G,7)
(F,4) (H,3) (I,5)

- void f3() – Perform breadth-first traversal from the root and delete by copying the first node having both 2 sons and price < 7. Output in the file **f3.txt** must be the following:

(C,8) (D,6) (E,9) (F,2) (G,7) (H,1) (I,3) (J,5) (K,4)
(C,8) (J,5) (E,9) (F,2) (G,7) (H,1) (I,3) (K,4)

- void f4() – Perform **breadth-first traversal** from the root and **find the first node p having left son and price < 7**. Rotate p to right about its' left son. Output in the file **f4.txt** must be the following:

(C,8) (D,6) (E,9) (F,2) (G,7) (H,1) (I,3) (J,5) (K,4)
(C,8) (F,2) (E,9) (H,1) (D,6) (I,3) (G,7) (J,5) (K,4)

Question 3: (2 marks)

In this question you should complete some methods in **Graph.java** file.

The class Graph is the implementation of a graph. The following methods should be completed:

- void f1() - Perform depth-first traversal (to the file f1.txt) from the vertex i=1 (the vertex B) but display vertices with their degrees in bracket. **Hint:** copy depth(...) to depth2(...) and modify the latter one. Content of the output file **f1.txt** must be:

B G A E F I C H D

B(1) G(2) A(4) E(3) F(3) I(3) C(1) H(2) D(1)

- void f2() – Apply the Dijkstra's shortest path algorithm to find the shortest path from the vertex 0 (A) to the vertex 4 (E). (Note that in the weighted matrix, the value 999 is considered as infinity). Write 2 lines into the file f6.txt. The first line contains the list of vertices in the shortest path. The second line contains shortest distances to the vertices in the first line. Content of the output file **f2.txt** must be:

A C F E

0 9 11 20

- void f3() – Supposed the given graph has Euler's cycle. Apply the pseudocode in the Graph.java file to write statements to find the Euler's cycle from the vertex 1 (B). Output in the file **f3.txt** must be the following:

B D E D C B E G F A B