

University of Newcastle
Discipline of Computer Science and Software Engineering
Semester 2, 2015 - SENG1120/6120

Assignment 3

Due using the Blackboard Turnitin submission facility:
23:59PM – 6 November 2015

NOTE: *The important information about submission and code specifics at the end of this assignment specification.*

In lectures we have discussed the use of binary trees. This assignment will require the implementation of a binary tree and several algorithms for element insertion, removal, search and tree traversal.

Assignment Task

Your task in this Assignment is to implement a binary search tree data structure (BSTree), which will store generic objects, i.e. use templates. BSTree will use a class named BTreeNode, for binary-tree node.

The type of object used in this assignment is called MechPart and it should store two data values: `string code` and `int quantity`. The data structure should implement functions to add, remove and search/modify an element, using the code as the search string, as shown in the lectures.

Your task is to implement the backend of a database for a maintenance company. The company keeps an inventory of mechanical parts and wants to store the information in the data structure developed by you.

Initially, you should create an interface class named DBInterface, which is menu-based system to interface with the user, with the following options:

- 1) Read current inventory file:** Will query the user for a filename, read it and populate a binary search tree.
- 2) Read/Exec monthly update file:** Will query the user for a filename, read it, and modify the tree by adding/removing mechanical parts. Remember that if a part has `quantity = 0`, the corresponding node should be removed.

Hint: When a node is to be deleted, there are two options:

- i. Left child is empty: Replace the node by its right child.
- ii. Left child is not empty: Replace the node by the highest element in the left branch. (Check the textbook, page 1373)

3) Display statistics: Displays the number of different parts and the total number of parts.

4) Search list items file: Will query the user for a filename, search for the parts listed in it and display their information. It will also present the average number of comparisons to find an item (i.e. total number of comparisons/number of items searched). If an item is not found, the quantity displayed should be zero.

5) Quit: Exit the program.

`DBInterface` will instantiate a binary tree and generate the calls to modify the contents of the database. In addition, you will create a `.cpp` named `DBDemo` that will contain the `int main()` function and simply instantiate and call `DBInterface`, passing control to it.

Useful steps for testing your implementation:

- 1) Read the current inventory file (menu option 1), which contains a list of parts codes and quantities; and populate a binary search tree with it. You can populate the tree in the same sequence as the parts are read from the file.
- 2) Display the statistics of the inventory (menu option 3).
- 3) Read the monthly inventory update file (menu option 2). The file will contain parts codes and positive/negative numbers. A positive number indicates that the company increased the inventory of that item. A negative number indicates the inventory decreased.
- 4) Display the statistics of the inventory and check if the results are consistent (menu option 3).
- 5) Read a third file with parts codes to search (menu option 4).

Please remember that incorrectly submitted assignments will not be marked.

- Assignments that do not use the specified class names **will not be further marked.**
- Assignments without a completed Cover Sheet **will not be further marked.**

You should provide all your `.h`, `.cpp`, `.template` files, and a `Makefile`. Also, if necessary, provide a `readme.txt` file containing any instructions for the marker. Each program file should have a proper header section including your name, course and student number, and your code should be properly documented with inline comments. **Remember, a completed Assignment Cover Sheet should accompany your submission.**

- Compress all your files, including the cover sheet, into a single `.zip` file and submit it in by clicking in a link that I will create in the Assignments section on Blackboard especially for that.

Remember that this assignment will take many hours to complete, and the submission itself should take less than 15 minutes. **Do not waste all your effort by submitting it incorrectly.**

Remember that your code should compile and run correctly using GNU C++. Test it using Cygwin installed in the computers in ES409.

Late submissions are subject to the rules specified in the Course Outline.
This assignment is worth 15 marks of your final result for the course.