Binary Search Trees

Description In this lab your goal is to implement standard operations of **binary search trees**, including insert and delete. See section 12.3 in the textbook. A sample class structure, with empty functions, is given in the support code. You can either use the given class structure or create your own. In this assignment the keys are integers. Your code will be tested for examples consisting of distinct keys. You will use Grade05 to test your code. Your execution file name must be "BST.exe". Refer to the previous lab assignments for instructions on how to use the grading tool.

In the input, each starts with 'e', 'i', 'd', 'oin','opre', or 'opost'. For each line, you will have to do the following.

- i(key): Insert (key) into the BST. For example, i2 implies "Insert key 2 into the BST."
- d(key): delete (key) from the BST. For example, d2 implies "delete key 2 from the BST." Do nothing if the BST doesn't have the key.
- opre: output all keys via preorder walk.
- opost: output all keys via postorder walk.
- oin: output all keys via inorder walk.
- e: finish your program.

Example input and output

The following example shows an execution of the program in interactive mode. See the input files and output files under the testfiles folder for examples where input and output are separated.

```
i3
i1
i5
i7
oin
1
3
5
7
d7
oin
1
3
5
opre
3
1
5
```

```
opost
1
5
3
e
```

Submission As usual, before the posted deadline, submit a .zip or zipped tar archive of your program through the assignments page of CatCourses. Please use your UCMNetID as the filename for the zipped archive. Be careful since CatCourses strictly enforces the assignment deadline. Recall that submission alone is not enough, you must present your work to your TA before submission.

Important Reminder Never change the grading scripts of the files under the "testfiles" folder. If you do so, it will be considered as SERIOUS CHEATING.