Practical Test : IB Computer Science HL Binary Search Trees + Objects

Name: _____ Date: 21/09/2017

Binary Search Trees with Objects

Given the files provided to you (BNode.java, Student.java, StudentTest.java, and student_data.txt), do the following:

- ✓ modify **BNode.java** so that the data held by the node is a *Student* object.
- ✓ write an ***insertByHeight*** method so that the data is added to the (*treeH*) binary search tree using student *ID* as the key field/attribute
- ✓ write an ***insertByName*** method so that the data is inserted in the (*treeName*) binary search tree using student *name* as the key field/attribute
- ✓ write methods to print trees in ***pre-order***, ***post-order*** and ***in-order*** traversals
- ✓ write methods to print trees in ***descending*** order
- ✓ use the four traversal methods on the ***treeH*** binary search tree…
- ✓ …as well as the ***treeName*** BSTs.

Work through the test from the beginning. Your program should build and grow –do not start a new program for each point.

During this test you may use **any** resources that you have created, class resources and notes but you may **not** use Internet.

<<< Please Turn Over >>>

Practical Test : IB Computer Science HL Binary Search Trees + Objects

Instructions	*Program Display/Screen Output*
1. Modify **BNode.java** so that the data held by the node is a *Student* object.	
2. Modify **BNode.java** so that the node can be accessed and its data can be easily printed/output/displayed.	
3. Add the root node to the binary search tree	`* Building treeH with root value 66555: KHAN Genghis @ 2.12 m tall`

4. Display the contents of ***treeH*** on the screen as they are inserted into the binary tree to prove that your insert method works properly (adding nodes to the left and right)	`* Building treeH with root value 66555: KHAN Genghis @ 2.12 m tall`
	`< Inserted 63444: BANNER Saga @ 1.97 m tall to left of node 66555: KHAN Genghis @ 2.12 m tall`
	`< Inserted 35444: MOORE Michael @ 1.72 m tall to left of node 63444: BANNER Saga @ 1.97 m tall`
	`> Inserted 57000: PATAN Doug @ 1.77 m tall to right of node 35444: MOORE Michael @ 1.72 m tall`
	`> Inserted 64999: HELL Boy @ 2.21 m tall to right of node 66555: KHAN Genghis @ 2.12 m tall`
	`> Inserted 60222: IVO Anthony @ 1.92 m tall to right of node 57000: PATAN Doug @ 1.77 m tall`
	`< Inserted 57333: WILSON Slade @ 1.9 m tall to left of node 60222: IVO Anthony @ 1.92 m tall`

5. Display the contents of ***treeName*** on the screen as they are inserted into the binary tree to prove that your insert method works properly (adding nodes to the left and right)	`* Building treeName with root value 66555: KHAN Genghis @ 2.12 m tall`
	`< Inserted 63444: BANNER Saga @ 1.97 m tall to left of node 66555: KHAN Genghis @ 2.12 m tall`
	`> Inserted 35444: MOORE Michael @ 1.72 m tall to right of node 66555: KHAN Genghis @ 2.12 m tall`
	`> Inserted 57000: PATAN Doug @ 1.77 m tall to right of node 35444: MOORE Michael @ 1.72 m tall`
	`> Inserted 64999: HELL Boy @ 2.21 m tall to right of node 63444: BANNER Saga @ 1.97 m tall`
	`> Inserted 60222: IVO Anthony @ 1.92 m tall to right of node 64999: HELL Boy @ 2.21 m tall`
	`> Inserted 57333: WILSON Slade @ 1.9 m tall to right of node 57000: PATAN Doug @ 1.77 m tall`

6. Display the contents of ***treeH*** on the screen in **Pre-order**.	`BST by HEIGHT ========`
	`*pre-order:`
	`66555: KHAN Genghis @ 2.12 m tall`
	`63444: BANNER Saga @ 1.97 m tall`
	`35444: MOORE Michael @ 1.72 m tall`
	`57000: PATAN Doug @ 1.77 m tall`
	`60222: IVO Anthony @ 1.92 m tall`
	`57333: WILSON Slade @ 1.9 m tall`
	`64999: HELL Boy @ 2.21 m tall`

7. Display the contents of ***treeH*** on the screen in **In-order**.	`*in-order:`
	`35444: MOORE Michael @ 1.72 m tall`
	`57000: PATAN Doug @ 1.77 m tall`
	`57333: WILSON Slade @ 1.9 m tall`
	`60222: IVO Anthony @ 1.92 m tall`
	`63444: BANNER Saga @ 1.97 m tall`
	`66555: KHAN Genghis @ 2.12 m tall`
	`64999: HELL Boy @ 2.21 m tall`

8. Display the contents of ***treeH*** on the screen in **Post-order**.	`*post-order:`
	`57333: WILSON Slade @ 1.9 m tall`
	`60222: IVO Anthony @ 1.92 m tall`
	`57000: PATAN Doug @ 1.77 m tall`
	`35444: MOORE Michael @ 1.72 m tall`
	`63444: BANNER Saga @ 1.97 m tall`
	`64999: HELL Boy @ 2.21 m tall`
	`66555: KHAN Genghis @ 2.12 m tall`

9. Display the contents of ***treeH*** on the screen in **descending order**.	`*descending order:`
	`64999: HELL Boy @ 2.21 m tall`
	`66555: KHAN Genghis @ 2.12 m tall`
	`63444: BANNER Saga @ 1.97 m tall`
	`60222: IVO Anthony @ 1.92 m tall`
	`57333: WILSON Slade @ 1.9 m tall`
	`57000: PATAN Doug @ 1.77 m tall`
	`35444: MOORE Michael @ 1.72 m tall`

10. Ensure that all four traversal methods also work when used with the ***treeName*** BST.	**BST by NAME ======** ***pre-order:** **66555: KHAN Genghis @ 2.12 m tall** **63444: BANNER Saga @ 1.97 m tall** **64999: HELL Boy @ 2.21 m tall** **60222: IVO Anthony @ 1.92 m tall** **35444: MOORE Michael @ 1.72 m tall** **57000: PATAN Doug @ 1.77 m tall** **57333: WILSON Slade @ 1.9 m tall** ***in-order:** **63444: BANNER Saga @ 1.97 m tall** **64999: HELL Boy @ 2.21 m tall** **60222: IVO Anthony @ 1.92 m tall** **66555: KHAN Genghis @ 2.12 m tall** **35444: MOORE Michael @ 1.72 m tall** **57000: PATAN Doug @ 1.77 m tall** **57333: WILSON Slade @ 1.9 m tall** ***post-order:** **60222: IVO Anthony @ 1.92 m tall** **64999: HELL Boy @ 2.21 m tall** **63444: BANNER Saga @ 1.97 m tall** **57333: WILSON Slade @ 1.9 m tall** **57000: PATAN Doug @ 1.77 m tall** **35444: MOORE Michael @ 1.72 m tall** **66555: KHAN Genghis @ 2.12 m tall** ***descending order:** **57333: WILSON Slade @ 1.9 m tall** **57000: PATAN Doug @ 1.77 m tall** **35444: MOORE Michael @ 1.72 m tall** **66555: KHAN Genghis @ 2.12 m tall** **60222: IVO Anthony @ 1.92 m tall** **64999: HELL Boy @ 2.21 m tall** **63444: BANNER Saga @ 1.97 m tall**