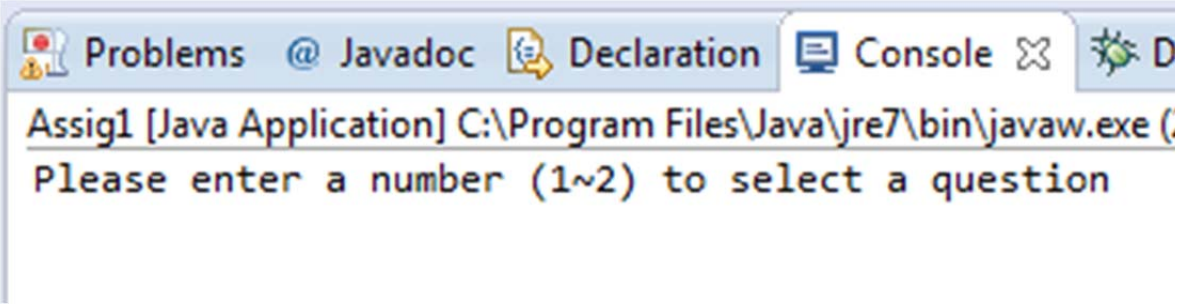
CSCI2010 Principle of Computer Science

Laboratory Night

Activity 0: Preparation

1. Create a project called “lab9” under Eclipse;
2. Copy “lab9.java” into project;
3. Copy “questions.java” into project;

Make sure your program can run correctly. An expected output of the UI is shown below.



```
Assig1 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (
Please enter a number (1~2) to select a question
```

Activity one: Binary Tree Height (5 marks)

Add *BTNode.java* and *LinkedBinaryTree.java* into lab9 project, these classes construct a linked binary tree. Add a method to the *LinkedBinaryTree* class that returns the height of a tree. Like many tree methods, this will require a fairly simple method in *LinkedBinaryTree* and another method in BTNode that does most of the work.

First, check to see if the root is null. If it is, then the tree is obviously empty and return 0. If it is not null, return *root.height*().

In *BinaryTreeNode*, add a method called height that uses the following algorithm:

1. If the node is not a leaf, store 2 values, *leftHeight* = *left.height*() and *rightHeight* = *right.height*()
2. Return the larger of the two values, plus 1.
3. If the node is a leaf, return 0.

Activity two: Spinning Trees (5 marks)

Add a method to the *LinkedBinaryTree* class that "*spins*" the tree.

To spin the tree, swap the children of every node in the tree. If done properly, this will reverse the order that elements are visited in an inorder traversal. Once again, add a simple method to *LinkedBinaryTree* to check to make sure root itself is not null then call a *BTNode* method to do all the work.

Once you've written the methods, make a tree with whatever type of objects you want and take it for a spin to test your work.

What need to be submitted?

Please submit the following java files,

- BTNode.java
- LinkedBinaryTree.java

=============================The End of Lab 9============================