CS500 Programming Assignment #5

Instructions:

• Due date: 11:59pm 2015 Nov 22nd.

• You must work individual for this assignment.

• For all programming assignments, please submit just the softcopy; please zip all source files and submit it WIU ONLINE. How to export source files from ECLIPSE to a zip file? Please check **Programming Assignment Submission Instructions** for details.

• Name your file as this rule: PROG#_SURNAME_FIRSTNAME.{zip|tar}”. E.g.“PROG1_John_Doe.zip”.
• In order to get full credit for the source code, your code must have in-line comments, must compile, and must be able to pass the test by instructor.

• Late policy: Late pass apply, after that, 10% penalty per day. No late submission later than 7 days after due day.

Exercise 1: **Simplifying Path (25 points):** Please write a program to simplify a given absolute path for a file (Unix-style) and then write a test program.

Example:

```
path = "/a/./b/../../c/", => "/c"
path = "/../", => "/"
```

Exercise 2: **Checking for Balanced Braces (25 points):** Please verify whether a string contains balanced braces, i.e. the braces should be closed in the correct order. Suppose the string only has one of the following braces: “(”, “)”,“[”,“]”, “{”,“}”.

```
String a  = “(){[()]}”           ->    balanced
String b = “()[}{]”, => "/"    ->    unbalanced
String c = “([)]{}”, => "/"    ->    unbalanced
```

Exercise 3: **Implement Queue using Stacks (25 points)**: Please implement the following operations of a queue using stacks, and then write a test program.

- push(x) -- Push element x to the back of queue.
- pop() -- Removes the element from in front of queue.
- peek() -- Get the front element.
- empty() -- Return whether the queue is empty.

Note: You must use only standard operations of a stack, i.e. push, pop, peek, empty

Exercise 4: CheckPalindrome(25 points): You have implemented LinkList Class for your Prog#4 Exercise#4, which should have insert and remove methods implemented. Now you need to implement one more method: checkPalindrome(), which is to check if the LinkList is a palindrome or not. Return true if the linklist is palindrome; otherwise, return false.

Example:

```
1-> 2-> 3 -> 2 -> 1 ==>     true
1-> 2-> 3 -> 2      ==>     false
```

Please use the following program to test your code.

```
public static void main( String [] args) {

     LinkList list = new LinkList();
      list.insert(0,new IntegerNode(10));
      list.insert(0,new IntegerNode(20));
      list.insert(0,new IntegerNode(10));

      System.out.println(list);
      System.out.println(list. checkPalindrome());

      list.insert(1,new IntegerNode(30));

      System.out.println(list);
      System.out.println(list. checkPalindrome());

    }
```