CSc 2720 - Data Structures: Assignment 2

How to Submit:

Turn the .java file in Folder Assignment 2 in iCollege no later than 11:59 p.m. on 02/22/2018.

Notes:

1. This is a common Google and other tech companies interview question.
2. Make sure not to copy the code from the internet. Turnitin (plagiarism software), will be activated for detecting plagiarism.

Refresher:

We have seen in class that a queue is an abstract data type, also called a First-In-First-Out (FIFO) data structure because the first element added/enqueued to the queue is always the first one to be removed/dequeued. A stack is an abstract data type, also called a Last-In-First-Out (LIFO) data structure because the first element added/pushed to the stack is always the last one to be removed/poped.

Requirements:

Write a new class called stackBasedQueue that implements following queue operations using only 2 stacks:

- enqueue(newItem) -- Add an integer input to the back of queue.
- dequeue() -- Remove the element from the front of the queue.
- peek() -- Get the front element.
- isEmpty() -- Return whether the queue is empty or not.
- stackBasedQueue() – Constructor that initialize the stacks to create a new queue object
import java.util.Stack;

public class stackBasedQueue{
    Stack<Integer> stack1;
    Stack<Integer> stack2;

    public void stackBasedQueue() {
        // Enter code Here
        // This is a stack initialization Stack<Integer> s = new Stack<Integer>();
    }

    public boolean isEmpty() {
        // Enter code Here
        return true;
    }

    public void enqueue(int newItem){
        // Enter code Here
    }

    public int dequeue() throws QueueException{
        // In case of problem:
        // throw new QueueException("You can not DEQUEUE on an EMPTY queue :((");
        // Enter code Here
        return 0;
    }

    public void dequeueAll(){
        // Enter code Here
    }

    public int peek() throws QueueException{
        // In case of problem:
        // throw new QueueException("You can not PEEK on an EMPTY queue :((");
        // Enter code Here
        return 1;
    }

    // Tester code
    public static void main(String[] args){
        stackBasedQueue q = new stackBasedQueue();
        q.enqueue(1);
        q.enqueue(2);
        q.enqueue(3);
        System.out.println("Ans#1: "+q.peek()); // Should be 1
        q.dequeue();
        q.dequeue();
        System.out.println("Ans#2: "+q.peek()); // Should be 3
        System.out.println("Ans#3: "+q.isEmpty()); // Should be false
        q.dequeueAll();
        System.out.println("Ans#4: "+q.isEmpty()); // Should be true
        System.out.println("Ans#5: "+q.peek());
        //Should throw exception with message: "You can not PEEK on an EMPTY queue :(( "
    }

    public class QueueException extends java.lang.RuntimeException{
        public QueueException(String s){
            super(s);
        }
    }
}
Expected Program output:

Exception in thread "main" QueueException: You can not PEEK on an EMPTY queue :(
    at stackBasedQueue.main(stackBasedQueue.java:19)

Ans#1: 1
Ans#2: 3
Ans#3: false
Ans#4: true