DSA Week 13 activities (Solutions)

This week, you are required to complete a questionnaire and two labs.

a. In this print out, answer all Week 13 questions.
b. Also, in this print out, complete Week 13 lab 1 & 2 using the lab computers.

Note: You can complete the activities in any order, however, make an effort to complete and understand everything which prepares you for well for test 2 & Final Exam.

DSA Week 13 Questions

1. What is a Queue?

A queue is a collection of objects that are inserted and removed according to the first-in, first-out (FIFO) principle.

A queue is a linear data structure that operates on the First-In, First-Out (FIFO) principle, where the first element added is the first one removed.

A queue is a data structure that can hold many elements.

2. Discuss the five methods used with queues.

1) **enqueue:** Adds a new element to the back of the queue
2) **dequeue:** Removes and returns the front element from the queue
3) **first:** returns the first element of the queue
4) **isEmpty:** Checks if the queue is empty.
5) **Size:** Finds the number of elements in the queue.

3. A queue has the elements (10, 28, 31).
    a. What happens to the queue when you enqueue (30)? (30, 10, 28, 31)
    b. What happens to the queue when you dequeue(30), dequeue (10), dequeue (28) and dequeue (31)? ()
    c. After performing the methods above, which elements are still in the queue? No
    d. After the three step above, if you run the method isEmpty() will it return true or false? Explain why it would return true or false. True, because the queue is empty or there are no elements in the queue.

4. Complete the sentence, Queues FIFO are implemented through Arrays and Linked List data structures.

5. What is the FIFO principle?

First-In-First-Out principle is a method or management of element/data where elements can be inserted at any time but only the element in the queue the longest can be removed next. Elements enter at the back and are removed from the front. For example: the metaphor of ATM line.

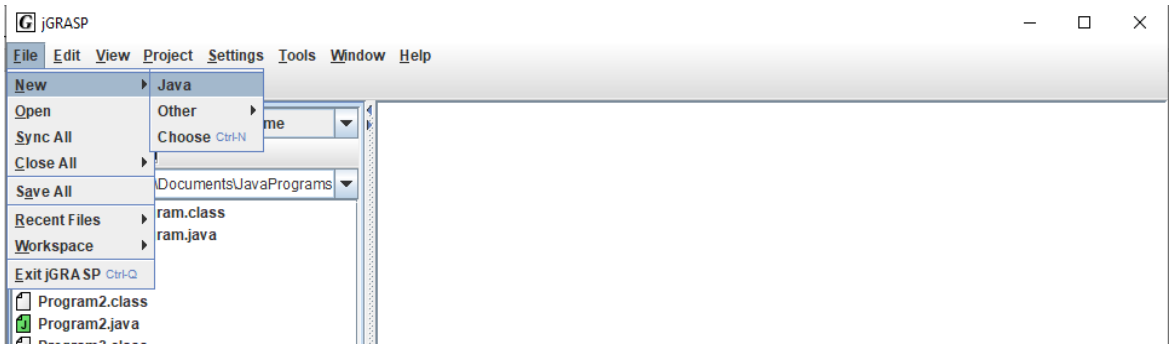
6. How to you differentiate between stacks and queue data structures?

First is their arrangement of data elements is different. Stacks follow the Last-In-First-Out principle where the last element is inserted and first element is removed, whereas, queue follow the First-In-First-Out principle where the first element is inserted and first element removed.

DSA Week 13 Lab Activity (Week9Lab1)

Using the lab computers create the following Java program using jGrasp!

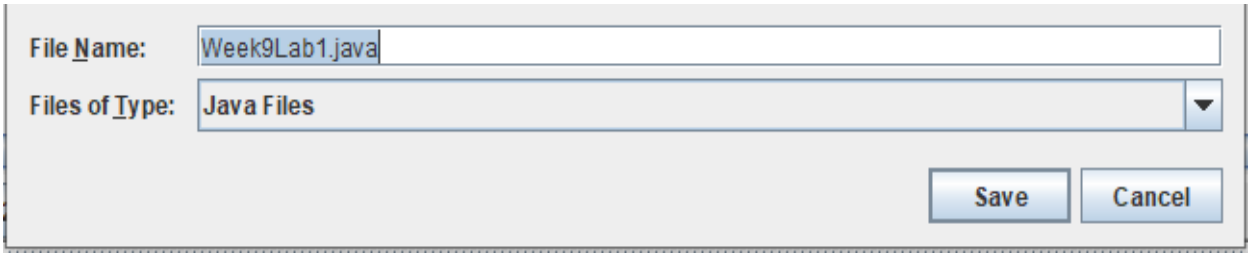
Step 1: Login to your lab computer and create a new java file in jGrasp.



Step 2: When the window below appears. Type the following code into jGrasp.

```
1  import java.util.LinkedList;
2  import java.util.Queue;
3
4  public class Week9Lab1 {
5
6     public static void main(String[] args) {
7        // Create a Queue object called itiCourses
8        Queue<String> itiCourses = new LinkedList<>();
9
10       // Enqueue elements into the queue
11       itiCourses.add("DIT");
12       itiCourses.add("DHRM");
13       itiCourses.add("DACC");
14       itiCourses.add("DICT");
15
16       // Peek or see the front element of the queue
17       System.out.println("Front element: " + itiCourses.peek());
18
19       // If queue is empty, print message; else print elements in the queue
20       if(itiCourses.isEmpty()) {
21          System.out.println("Queue is empty");
22       } else {
23          // Print the queue (it will display elements in FIFO order)
24          System.out.println("Queue contents: " + itiCourses);
25       }
26    }
27 }
```

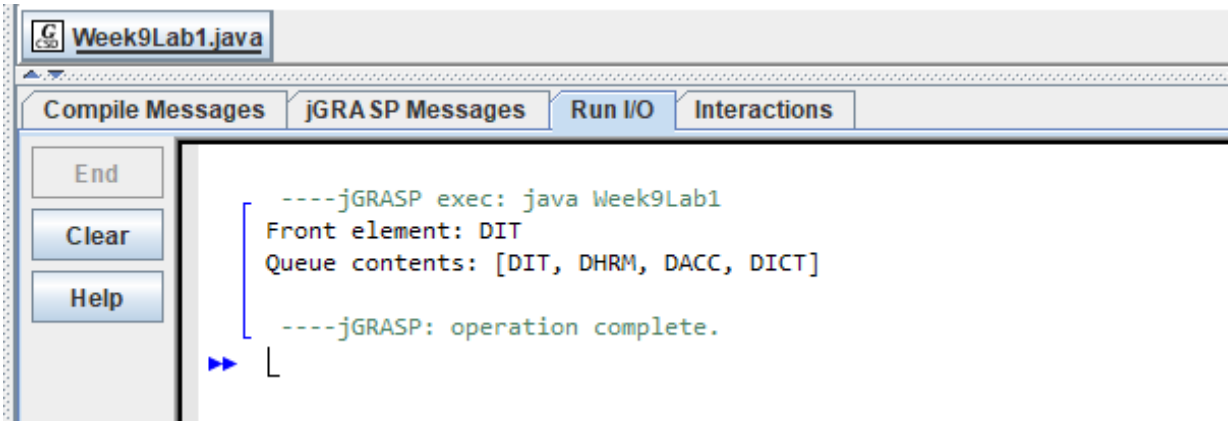
Step 3: Go to **file/save** to save your java program as **Week9Lab1**



Step 4: After saving, compile (**click on compile icon or on your keyword hold Ctrl + B**) to check for syntax errors.

Step 5: If compiling is successfully then run (**click on the find and run main method icon or on your keyboard hold Ctrl + R**) your program.

Step 6: If run is successful then you should see the following output in the console

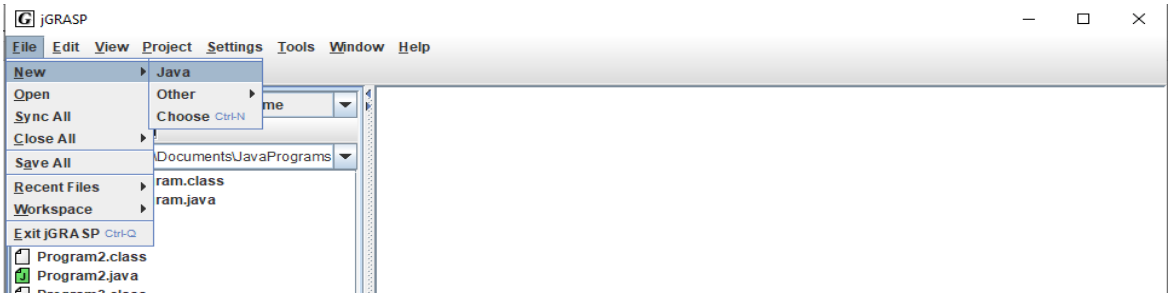


Step 7: Week9Lab1 Completed! Save your file for future Java lab activities.

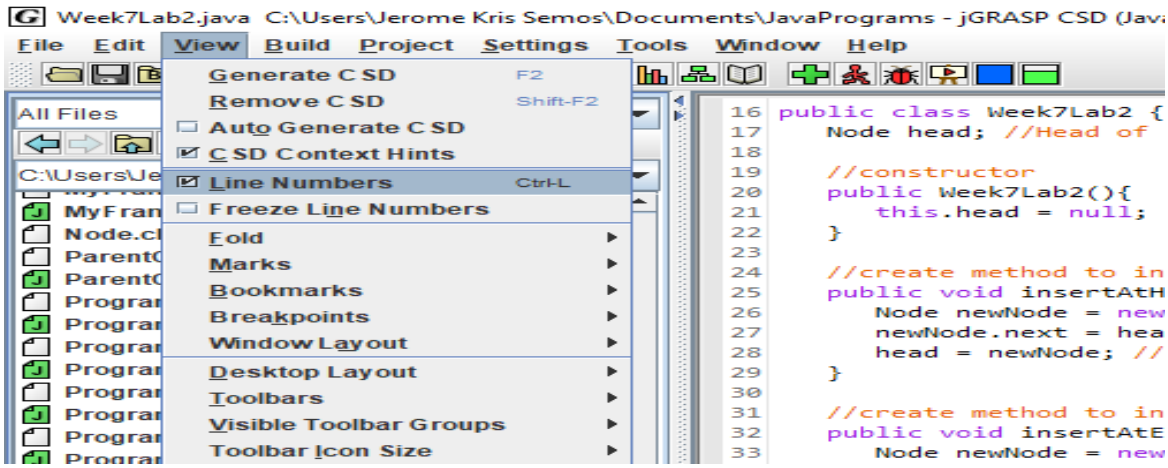
DSA Week 13 Lab Activity (Week9Lab2)

Using the lab computers create the following Java program using jGrasp!

Step 1: Login to your lab computer and create a new java file in jGrasp.



Step 2: Switch on line numbers, **Go to View\Line Numbers or hit Ctrl + L to enable line numbers.**

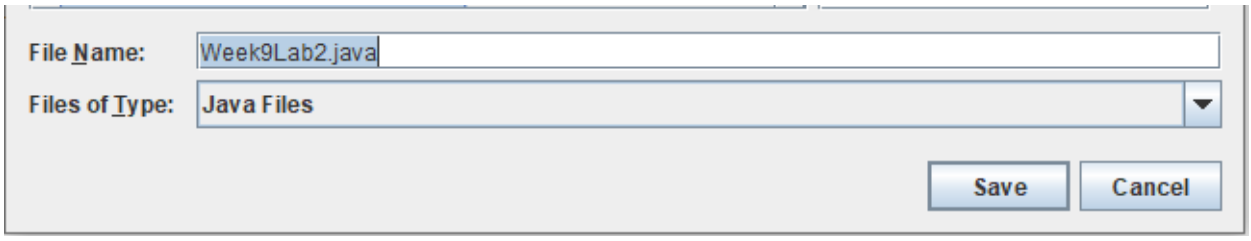


Step 3: Now, type the following code into jGrasp. **Note: this program has 87 lines of Java code, hence, we have to enable line numbers to assist our coding and resolve errors when coding, compiling or debugging.**

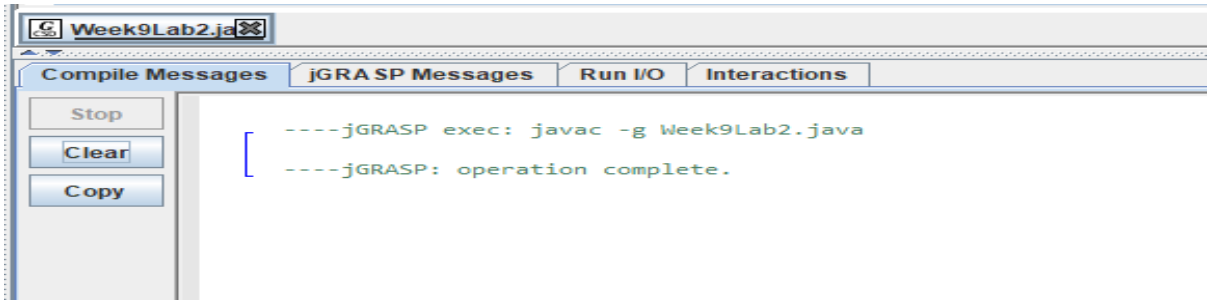
```java
import java.util.LinkedList;
import java.util.Queue;

public class Week9Lab2<E> {

    private Queue<E> queue; // Change Stack to Queue

    // Constructor
    public Week9Lab2() {
        this.queue = new LinkedList<>();
    }

    // Enqueue method (add elements to the queue)
    public void enqueue(E item) {
        queue.add(item);
    }

    // Dequeue method (remove elements from the front of the queue)
    public E dequeue() {
        if (!queue.isEmpty()) {
            return queue.poll();
        } else {
            System.out.println("Queue is empty.");
            return null;
        }
    }

    // Peek method (check front element)
    public E peek() {
        if (!queue.isEmpty()) {
            return queue.peek();
        } else {
            System.out.println("Queue is empty.");
            return null;
        }
    }

    // Check if queue is empty
    public boolean isEmpty() {
        return queue.isEmpty();
    }

    // Print queue elements
    public void printQueue() {
        if (queue.isEmpty()) {
            System.out.println("Queue is empty.");
        } else {
            System.out.println("Queue contents:");
            for (E item : queue) {
                System.out.println(item);
            }
        }
    }

```

```
55      public static void main(String[] args) {
56          // Create a generic queue for Strings
57          Week9Lab2<String> stringQueue = new Week9Lab2<>();
58          stringQueue.enqueue("DIT");
59          stringQueue.enqueue("DHRM");
60          stringQueue.enqueue("DACC");
61          stringQueue.enqueue("DICT");
62
63          System.out.println("String Queue:");
64          System.out.println("Front element: " + stringQueue.peek());
65          stringQueue.printQueue();
66
67          // Create a generic queue for Integers
68          Week9Lab2<Integer> intQueue = new Week9Lab2<>();
69          intQueue.enqueue(10);
70          intQueue.enqueue(20);
71          intQueue.enqueue(30);
72
73          System.out.println("\nInteger Queue:");
74          System.out.println("Front element: " + intQueue.peek());
75          intQueue.printQueue();
76
77          // Create a generic queue for Doubles
78          Week9Lab2<Double> doubleQueue = new Week9Lab2<>();
79          doubleQueue.enqueue(10.5);
80          doubleQueue.enqueue(20.75);
81          doubleQueue.enqueue(30.25);
82
83          System.out.println("\nDouble Queue:");
84          System.out.println("Front element: " + doubleQueue.peek());
85          doubleQueue.printQueue();
86      }
87  }
```

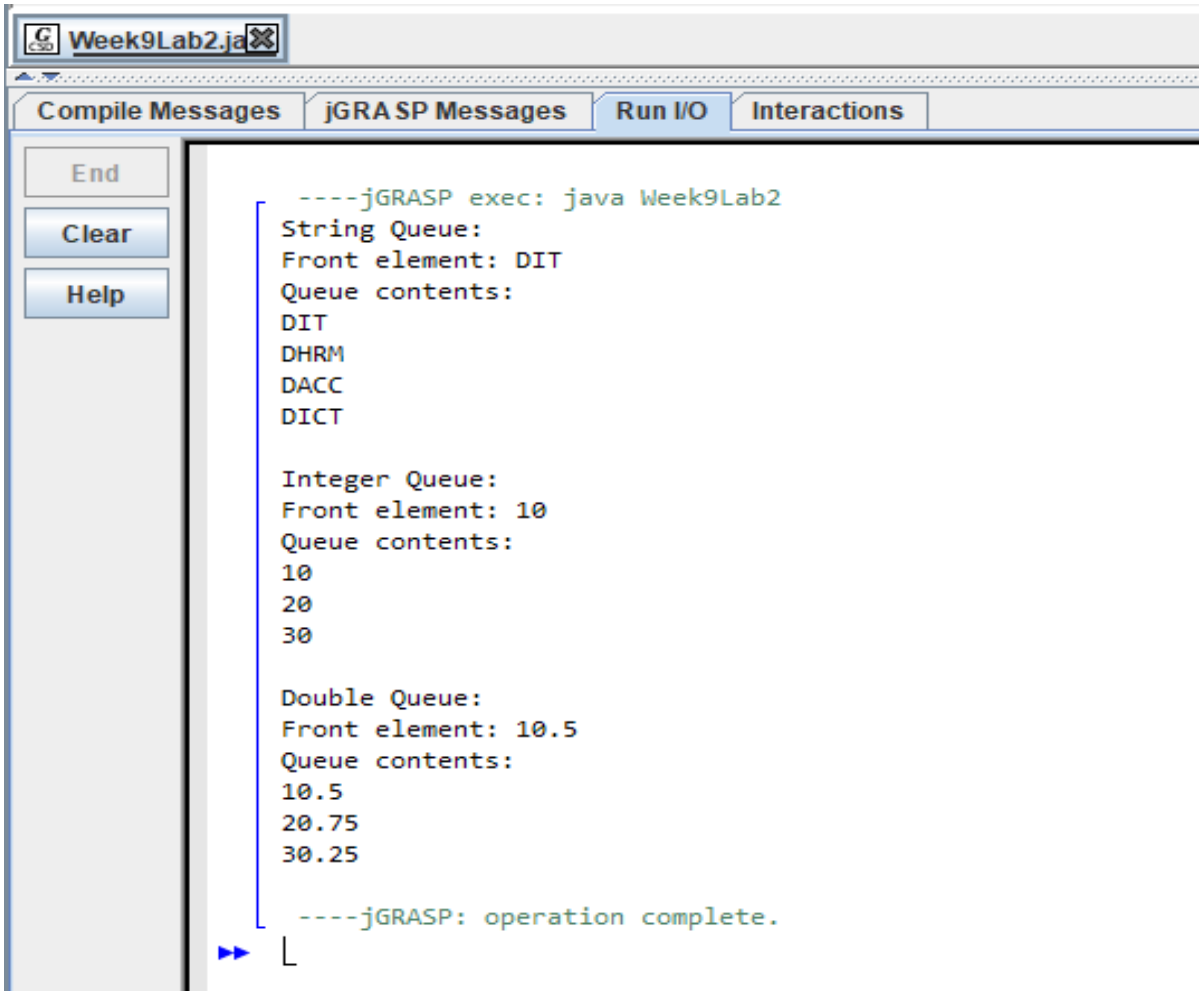
Step 4: After coding the program, go to **file/save** to save your java program as **Week9Lab2**



Step 5: After saving, compile (**click on compile icon or on your keyword hold Ctrl + B**) to check for syntax errors.



Step 6: If compiling is successfully then run (**click on the find and run main method icon or on your keyboard hold Ctrl + R**) your program.



Step 7: If successful your program should display an output like shown in the screenshot above.

Step 8: Week9Lab2 Completed! You have created your first generic Queue data structure program.

Summary of application of queues in the Real World

1. Using of task scheduling use queues to prior and order to receive tasks and complete them.
2. Operation system use queues to manage processes, requests and resources.
3. Computer networks use queues in networking protocols like TCP to manage packets transmitted over the network. Ensure delivery order and appropriate rate of network communication.