

DSA Week 7 activities

This week, you are required to complete two questionnaires and one lab.

- a. In your DSA textbook 1, answer questions 1, 2, 3, 4, 5, 9 and 13.
- b. In this print out, answer all week 7 questions.
- c. Also, in this print out, complete week 7 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, Major Assignment, Mid Semester Exam & Final Exam.

DSA Textbook 1: chapter review questions

1. Which of the following is not true? A reference to a class object
 - **b. has a size dependent on its class**
2. Access to the links in a linked list is usually through the **first** link.
3. When you create a reference to a link in a linked list, it
 - **d. can refer to any link you want**
4. How many references must you change to insert a link in the middle of a singly linked list? **2**
5. How many references must you change to insert a link at the end of a singly linked list? **1**
9. A double-end list
 - **a. can be accessed from either end.**
10. A special case often occurs for insertion and deletion routines when a list is **empty**.

DSA Week 7 Questions

1. What is a Linked List?

Any of the three answers are correct:

- A linked list class is a collection which can contain many objects of the same type like an ArrayList.
- A linked list is a collection of nodes that together form a linear ordering. The ordering is determined as in the child's game "follow the leader" to which each node is an object that stores a reference to an element and a reference called next, to another node.
- A linked list is a list where nodes are linked together. Each node contains data and a pointer or link.

2. List two advantages and disadvantages of using Linked List in programming.

Advantages: linked list have quick insertion and quick deletion of elements

Disadvantages: linked list are slow at searching & requires extra RAM or memory usage.

3. Referring to the Linked List data structure, what is a node?

Any of the two answers are correct:

- A basic unit of a Linked List containing data and a reference to the next node.
- Each element in a linked list that holds data and a link or pointer to next node is called a node.

4. Explain the two things or items that node stores.

Each node stores two things:

- i. data – the value or element of the node
- ii. next – a reference to the next node in the list.

5. What is RAM and how does it store a Linked List data structure?

RAM – Random Access Memory is where your variables, data types, data structures and instructions are stored on a computer. In computer, RAM is component that provides temporary storage for data and instructions that a processor needs to access.

Unlike Array, Linked list are store in non-contiguous or non-sequential memory locations. Meaning linked list nodes are scattered throughout memory locations.

6. Explain the difference between a singly linked list and doubly linked list.

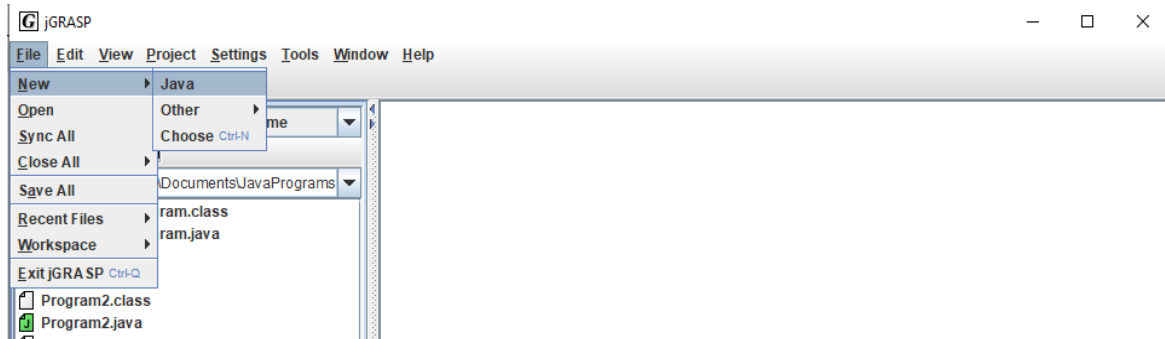
Singly Linked List: is a data structure consisting of a list where each node has a reference to the next node.

Doubly Linked List: is a data structure consisting of a list where each node has references to both the next and the previous node.

DSA Week 7 Lab Activity (Week7Lab1)

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.

```
/* DSA Week 7 Lab 1 */

import java.util.LinkedList; //import the LinkedList class or Java Package

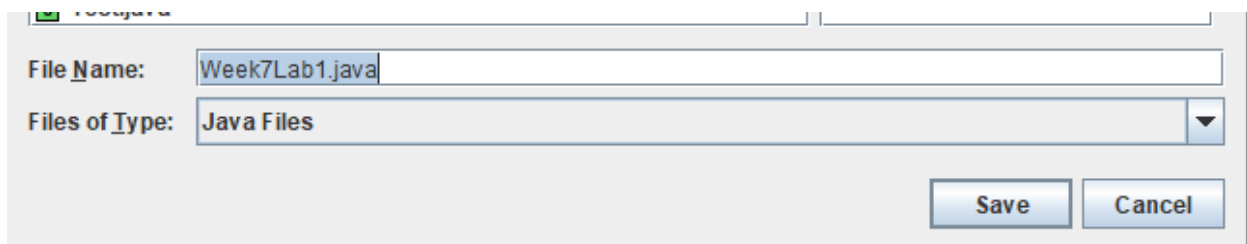
public class Week7Lab1 {

    public static void main(String[] args){
        //create Linked List object called itiCourses
        LinkedList<String> itiCourses = new LinkedList<String>();

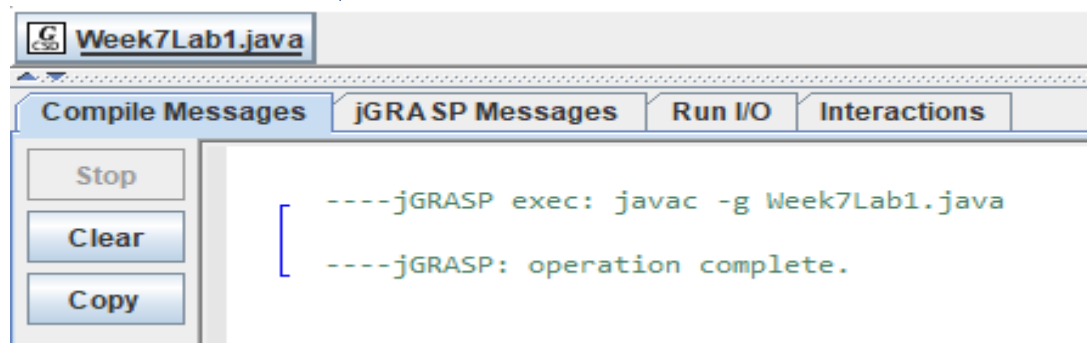
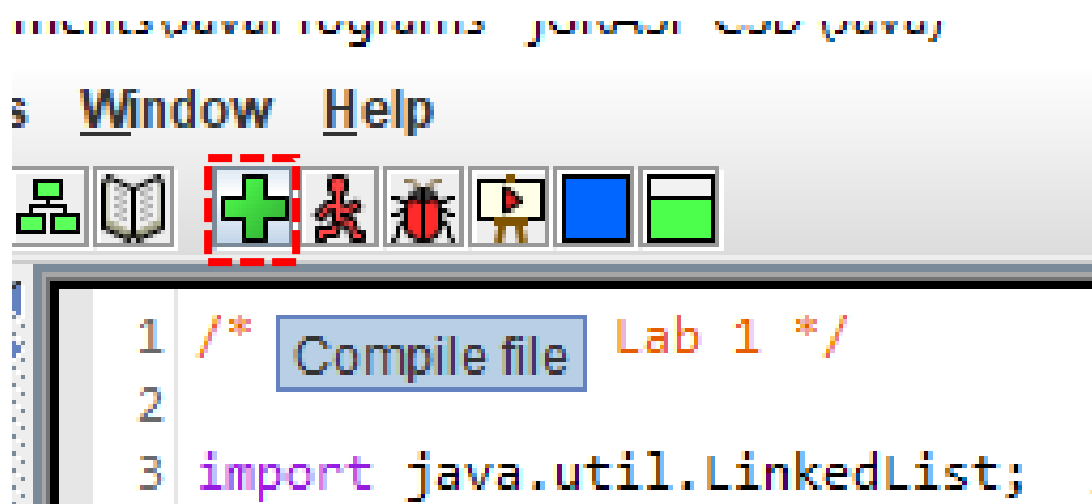
        //add nodes into the Linked List
        itiCourses.add("DIT");
        itiCourses.add("DHRM");
        itiCourses.add("DACC");
        itiCourses.add("DICT");

        //print to the console the Linked List
        System.out.println(itiCourses);
    }
}
```

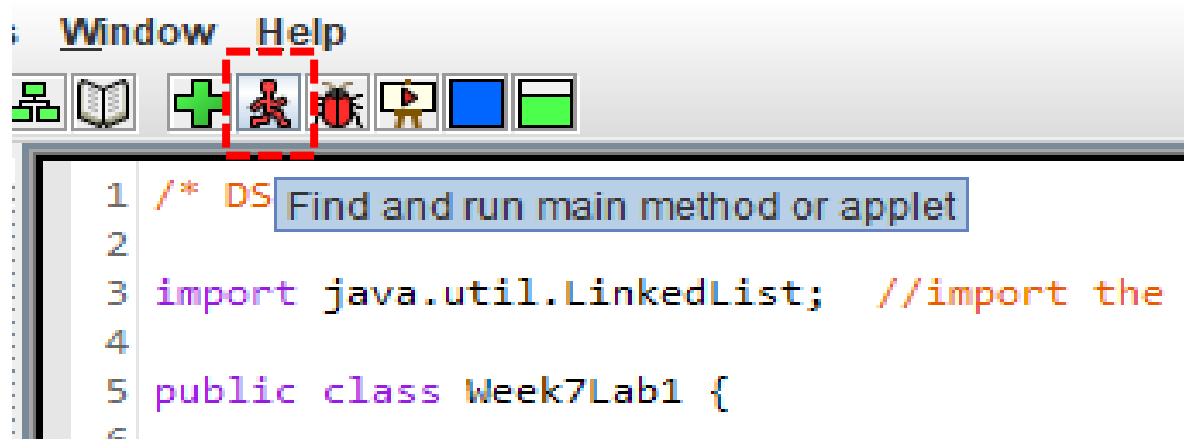
Step 3: Go to **file/save** to save your java program as **Week7Lab1**



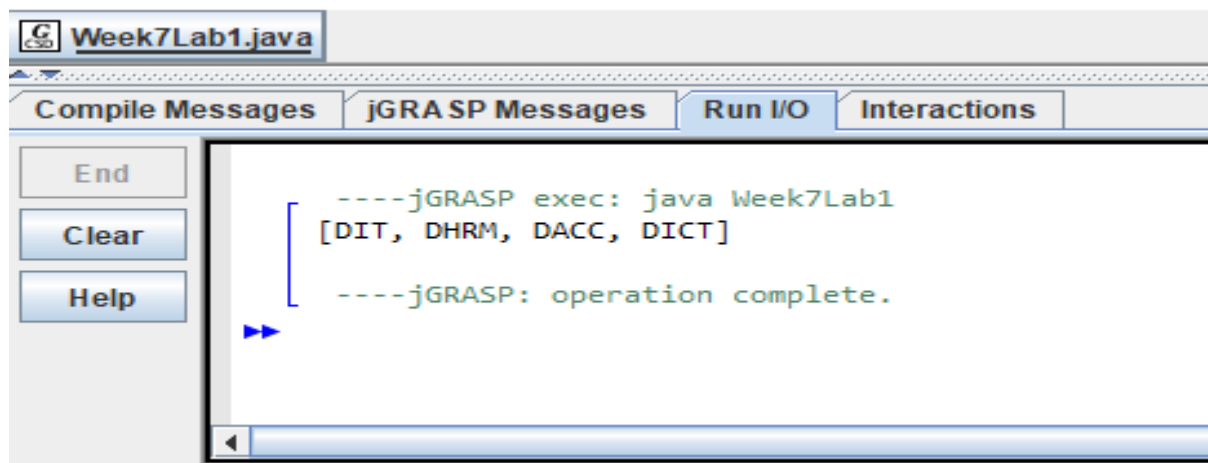
Step 4: After saving, compile (click on compile icon or on your keyboard hold Ctrl + B) to check for syntax errors.



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

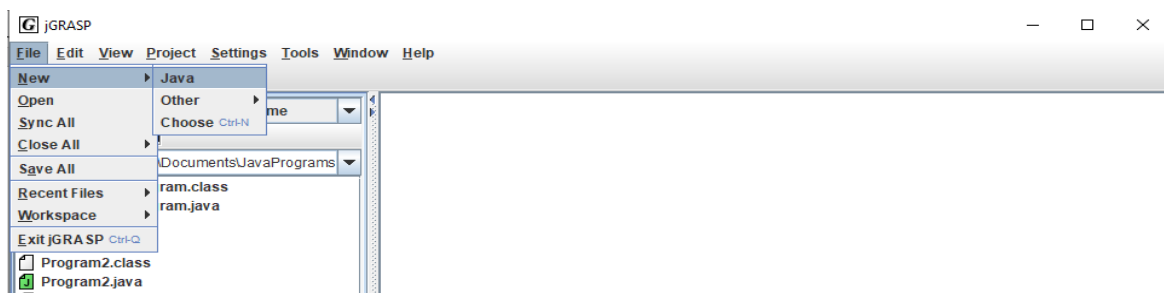


Step 7: Week7Lab1 Completed! Save your file for future Java lab activities.

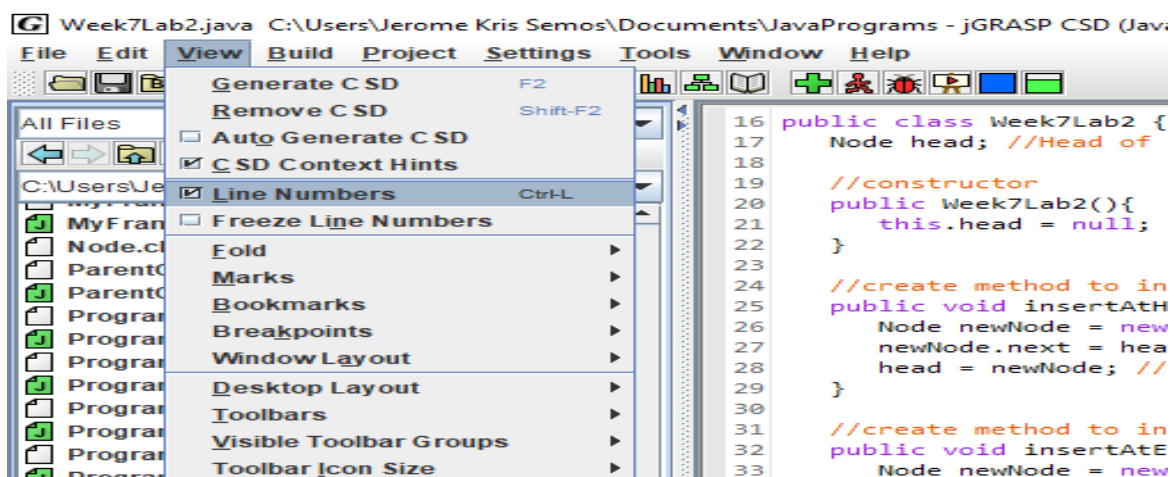
DSA Week 7 Lab Activity (Week7Lab2)

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 89 lines of Java code, hence, we have to enable line numbers to assist our coding and resolve errors when coding, compiling or debugging.

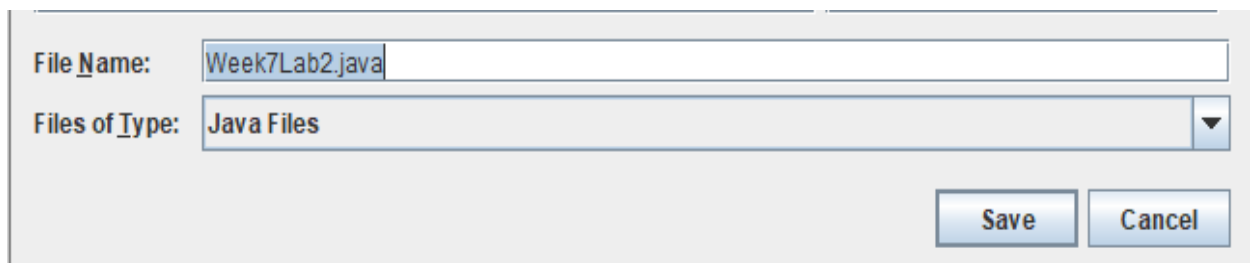
```
1  /* DSA Week 7 Lab 2 */
2
3  import java.util.LinkedList; //import the LinkedList class or Java Package
4
5  class Node{
6      int data;
7      Node next;
8
9      //constructor to create a new node
10     public Node(int data){
11         this.data = data;
12         this.next = null;
13     }
14 }
15
16 public class Week7Lab2 {
17     Node head; //Head of the linked list
18
19     //constructor
20     public Week7Lab2(){
21         this.head = null;
22     }
23
24     //create method to insert a node at the beginning
25     public void insertAtHead(int data){
26         Node newNode = new Node(data);
27         newNode.next = head; //new node points to the previous head
28         head = newNode; //head is updated to the new node
29     }
30
31     //create method to insert a node at the end
32     public void insertAtEnd(int data){
33         Node newNode = new Node(data);
34         if(head == null){
35             head = newNode; //if the list is empty, make new node the head
36             return;
37         }
38         Node temp = head;
39         while (temp.next != null){
40             temp = temp.next; //traverse until the last node
41         }
42         temp.next = newNode; //link the last node to the new node
43     }
44
45     //method to display the list
46     public void display(){
47         Node temp = head;
48         while (temp != null){
49             System.out.print(temp.data + "->");
50             temp = temp.next;
51         }
52         System.out.println("null"); //end of the linked list
53     }
54
55     //method to find the size of the linked list
56     public int size(){
57         int size = 0;
58         Node temp = head;
59         while(temp != null){
60             size++;
61             temp = temp.next;
62         }
63         return size;
64     }
65 }
```

```

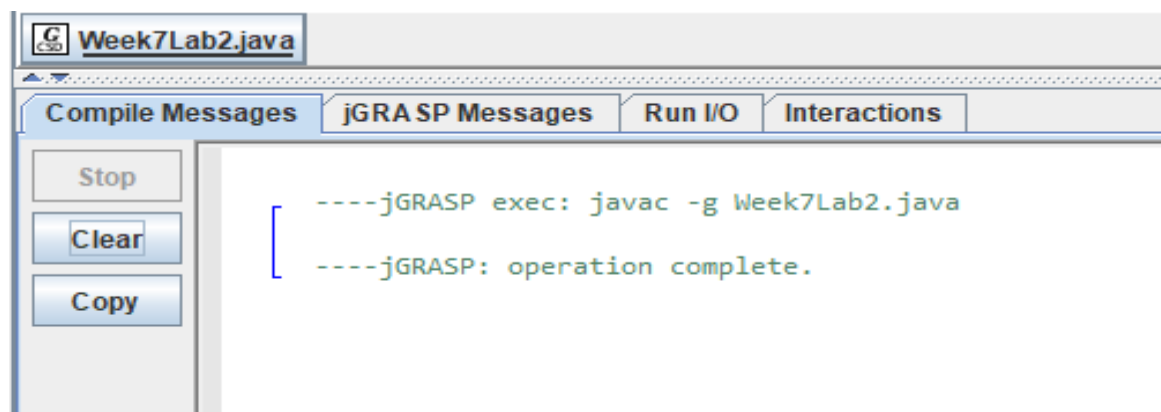
65
66 public static void main(String[] args){
67     //create Linked List object called itiCourses
68     Week7Lab2 myLinkedList = new Week7Lab2();
69
70     //add nodes at the head and end
71     myLinkedList.insertAtHead(940);
72     myLinkedList.insertAtHead(880);
73     myLinkedList.insertAtHead(830);
74     myLinkedList.insertAtHead(790);
75     myLinkedList.insertAtHead(750);
76     myLinkedList.insertAtEnd(660);
77     myLinkedList.insertAtEnd(650);
78     myLinkedList.insertAtEnd(590);
79     myLinkedList.insertAtEnd(510);
80     myLinkedList.insertAtEnd(440);
81
82     //print to the console the Linked List
83     System.out.println("Linked List after insertions:");
84     myLinkedList.display();
85
86     //print size of the list
87     System.out.println("Size of the list is: " + myLinkedList.size());
88 }
89 }

```

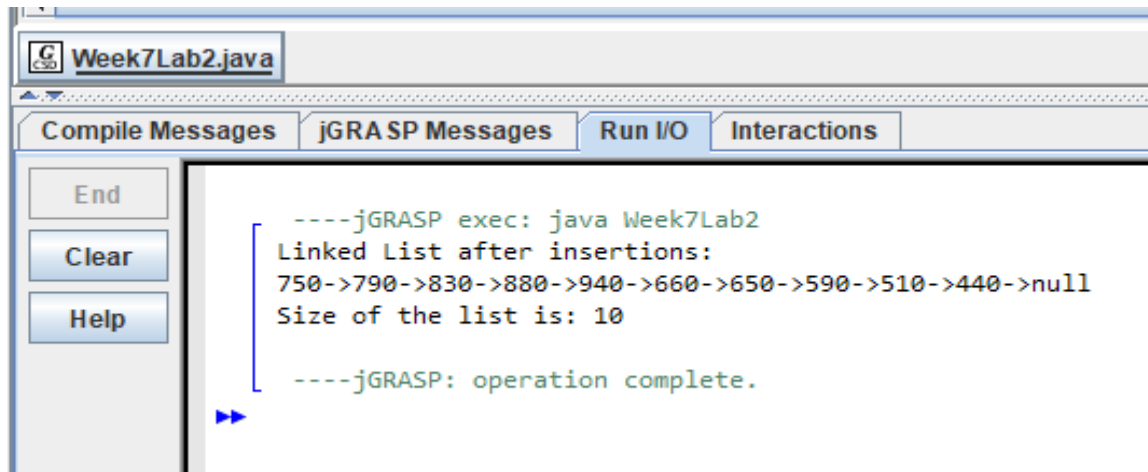
Step 4: After coding the program, go to **file/save** to save your java program as **Week7Lab2**



Step 5: After saving, compile (click on compile icon or on your keyboard 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: Week5Lab2 Completed! You have created your first Singly Linked List data structure program.

Summary of application of Linked List in the Real World

1. Music players use linked list to create playlists and play songs either from the beginning or end of the list. Each song is stored as a node in the linked list with the next link connecting to the next song on the playlist.
2. Web browsers use linked list to store the browsing history of the user, therefore, allowing users to navigate to previous links, search through history of URLS and sort history browser browsing over a predefined time range.