

INFO-C307 Laboratory Session #4

LAB GOALS

- Understanding Reference variables (what some languages call “Pointers”)
- Understanding the (Dot) notation.
- Manually creating a small linked list

Step 1: Create a new Java project called Lab4.

Step 2: Now create a new class called **aNode**.

```
class aNode {
    char data;
    aNode next;

    aNode(char mydata) { // Constructor
        data = mydata;
        next = null;
    }
};
```

Step 3: In the main() function of the driver class (Lab4), instantiate an object of type aNode and print the content of its class variables. (see code below)

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

    System.out.println("Create and initialize Node_1");
    aNode myNode_1;           // reference to an object of type "aNode"
    myNode_1 = new aNode('A'); // Instantiate an object of type aNode
    System.out.println(myNode_1.data);
    System.out.println(myNode_1.next);

}
```

The above is an example of **dynamically allocating an object** of type “aNode”. Once the object is created, we can use the “dot” notation to access and display its properties.

Step 4: Compile and run the program to make sure the above steps are working properly.

```
Create and initialize Node_1
A
null
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

Step 5: Now add the following code after the code in step 3 and note the syntactic difference for creating myNode_2. (two lines has been combined into one!)

```
System.out.println("Create and initialize Node_2");
aNode myNode_2 = new aNode('B'); // Alternate method of creating object
System.out.println(myNode_2.data);
System.out.println(myNode_2.next);
```

Step 6: Compile and run the program to make sure the above steps are working properly.

```
Create and initialize Node_1
A
null
Create and initialize Node_2
B
null
```

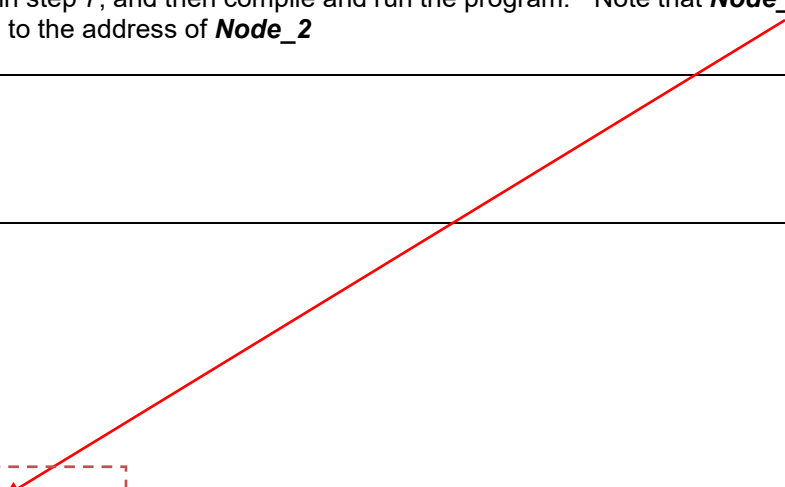
Step 7: Now add the following lines to the program. The second line in the code below connects **node_1.next** to **node_2**.

```
System.out.println("Connecting Node_1 to Node_2");
myNode_1.next = myNode_2;
```

Step 8: Place the following code below the code in step 7, and then compile and run the program. Note that **Node_1.next** is no longer "null". It has been changed to the address of **Node_2**

```
System.out.println(myNode_1.data);
System.out.println(myNode_1.next);
System.out.println(myNode_2.data);
System.out.println(myNode_2.next);
```

```
Run:
Create and initialize Node_1
A
null
Create and initialize Node_2
B
null
Connecting Node_1 to Node_2
A
dynamic_static_allocation.aNode@15db9742
B
null
BUILD SUCCESSFUL (total time: 0 seconds)
```



Step 9: Now let's try to understand pointers a little better. As you know, the above nodes are created in memory, and they have a physical address. So, let's try to find out what is the actual memory location for these **two nodes**.

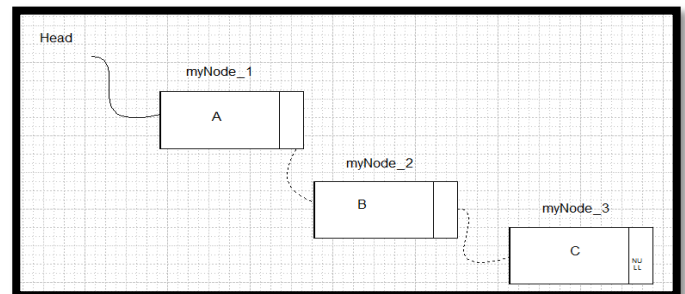
```
System.out.println(myNode_1);
System.out.println(myNode_2);

run.
Create and initialize Node_1
A
null
Create and initialize Node_2
B
null
Connecting Node_1 to Node_2
A
dynamic_static_allocation.aNode@15db9742
B
null
dynamic_static_allocation.aNode@6d06d69c
dynamic_static_allocation.aNode@15db9742
BUILD SUCCESSFUL (total time: 0 seconds)
```

Note that the address of **Node_2** is the same as the address of **Node_1.next**

Step 10: Now on your own, perform the following steps:

- 1) Create a reference variable called **Head** and connect it to **myNode_1**. (see image)
- 2) Create another node called **myNode_3**, instantiate it with the value '**C**' and connect **myNode_2.next** to **myNode_3**. (see image) (also see steps 5 and 7 to remind yourself)



Step 11: Now add the following code at the end of step 10, then compile and run your code. This code creates a temporary pointer to the head of the linked list, then uses a while loop to navigate the nodes in the linked list and print its internal data.

```
System.out.println("Printing the connected linked list (Node_1, Node_2, Node_3)");
aNode ptr;      // create a temporary reference variable
ptr = Head;
while (ptr != null) {
    System.out.println(ptr.data);
    ptr = ptr.next;
}
```

Create and initialize Node_1

A

null

Create and initialize Node_2

B

null

Connecting Node_1 to Node_2

A

dynamic_static_allocation.aNode@15db9742

B

null

dynamic_static_allocation.aNode@6d06d69c

dynamic_static_allocation.aNode@15db9742

Printing the connected linked list (Node_1, Node_2, Node_3)

A

B

C

BUILD SUCCESSFUL (total time: 0 seconds)

What to submit

- The final source code after step 11.
- Screen shot after step 11.