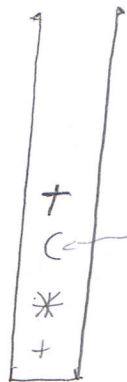


Cmp 338

①
9.15.09

2+3*4

(2+3)*4

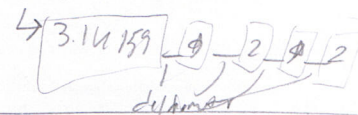


Stack

8+4*(2+3)*4+5

infix = "(" + infix + ")"

- 3.14159 * 2 * 2
- string tokenizer class
- delimiter



- Loops to find operators
- pre process string to remove spaces

token by token



```

if (c == '(')
{
    stack.push(c);
}
else
if (c == ')') (is(s.equals("(")))
{
    while (c != stack.isEmpty()) { stack.pop(); }
    postFix += stack.pop();
}
stack.pop();
else
{
    postFix += c;
}
return stack.top();
    
```

Changes

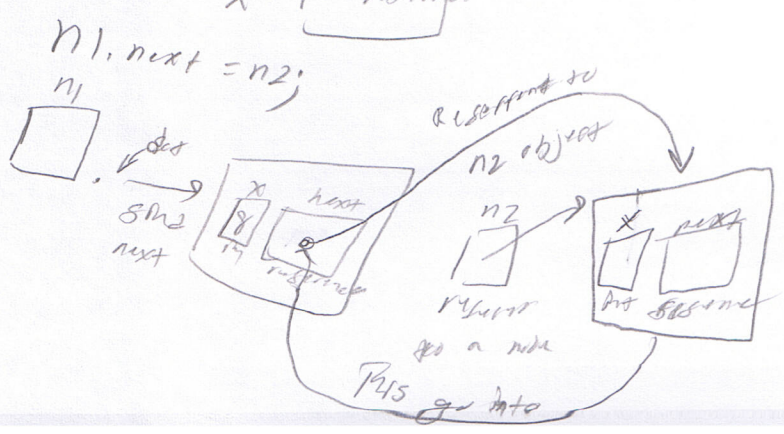
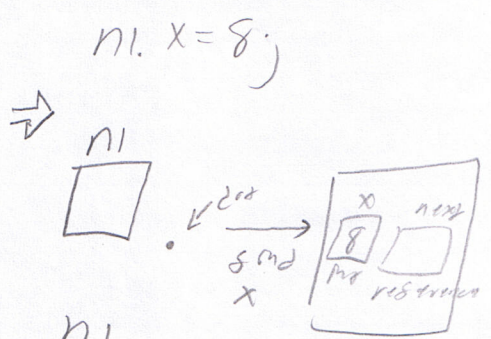
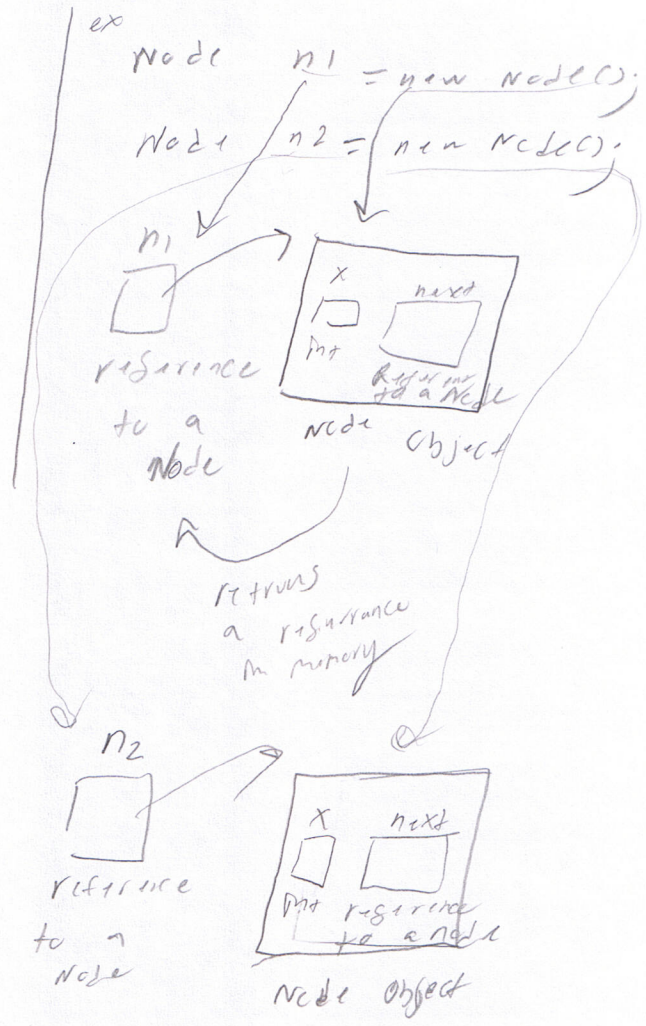
cmp 338

②

9.15.09

```

public class Node
{
    int x;
    Node next;
}
    
```

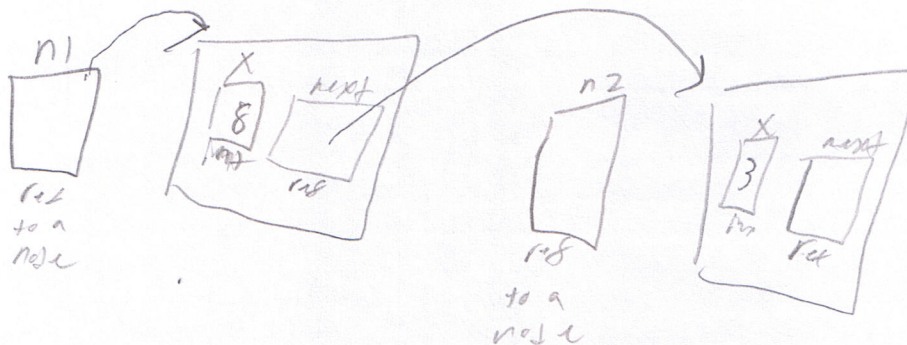


Comp 338

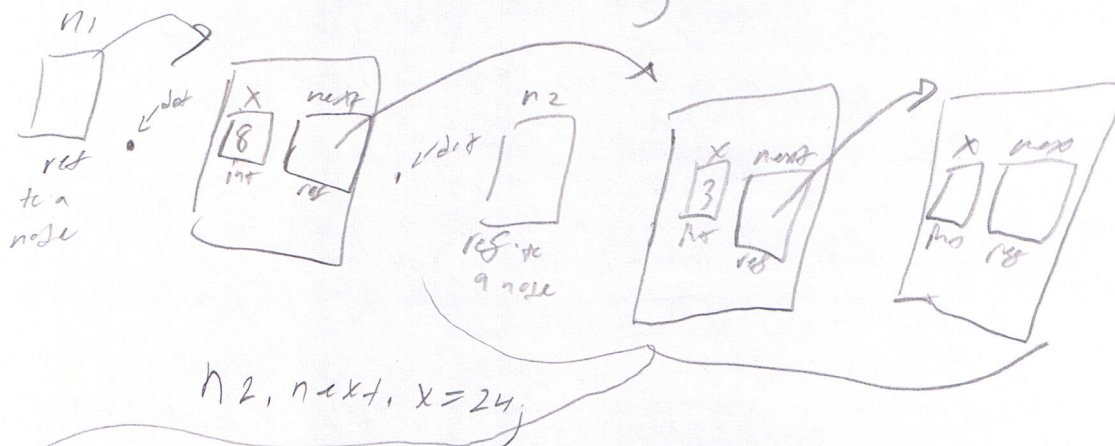
3
9.15.09

Linked List

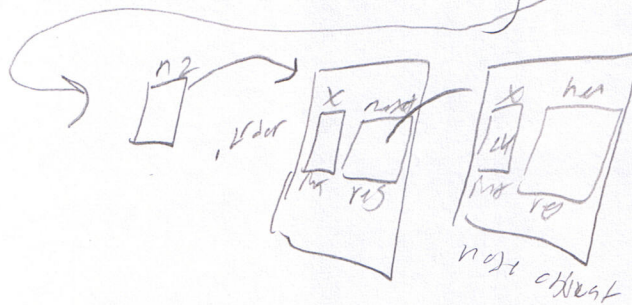
$n1, next, x = 3$



$n1, next, next = \text{new Node } ()$



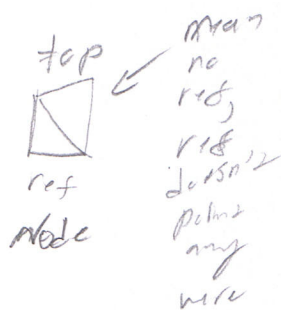
$n2, next, x = 24$



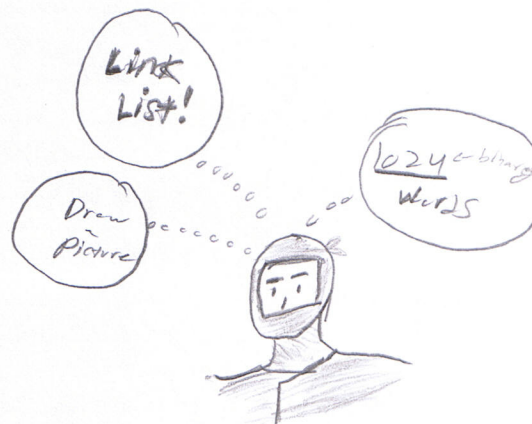
cmp 338

4

9.15.09



```
public class Node
{
    int x;
    Node next;
}
```



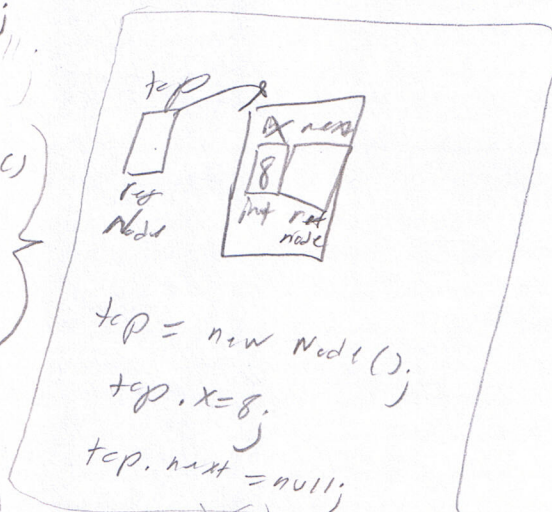
```
public class StackLLInt
{
```

```
    private Node top = null;
    private Node temp = null;
```

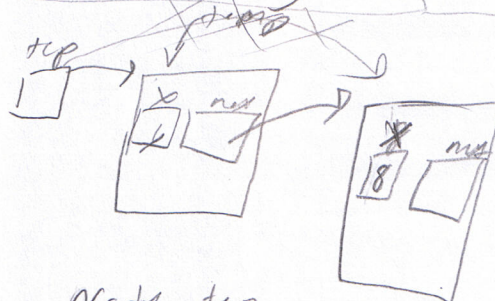
```
    public boolean isEmpty()
    {
        return top == null;
    }
```

```
    public void push(int x)
    {
```

```
        Node temp = new Node();
        temp.x = x;
        temp.next = top;
        top = temp;
    }
```



Stack.push(8);
Stack.push(6);



```
Node temp = new Node();
temp.x = 6;
temp.next = top;
top = temp;
```

