Program: Stack This!

Create a generic `Stack` class that supports the following operations (and only these) as defined in our lectures:

- Push: insert an item on top of the stack
- Pop: delete an item from the top of the stack (and return it)
- Peek: return (but do not delete) the item on top of the stack
- Size: return the size of the stack
- IsEmpty: return if the stack is empty
- IsFull: return if the stack is full
- Equals: compare two stacks
- Add: concatenate two stacks

Also include a constructor, a copy constructor, and the output capability – i.e.:

- `Stack()`
- `Stack(Stack s)`
- `String toString()`

Again, I am providing the `main` function and will test the functionality and implementation of your stack. You are to use it (unmodified) and build your generic stack on top of your generic list. As such, the only member variable your stack will *absolutely* require is a generic list. Implement your stack in a separate file (e.g., `Stack.java`) and make sure that `List.class` and `Node.class` are in the same directory.

So that we're all on the same page, please orient your stack so that its top is to the left.