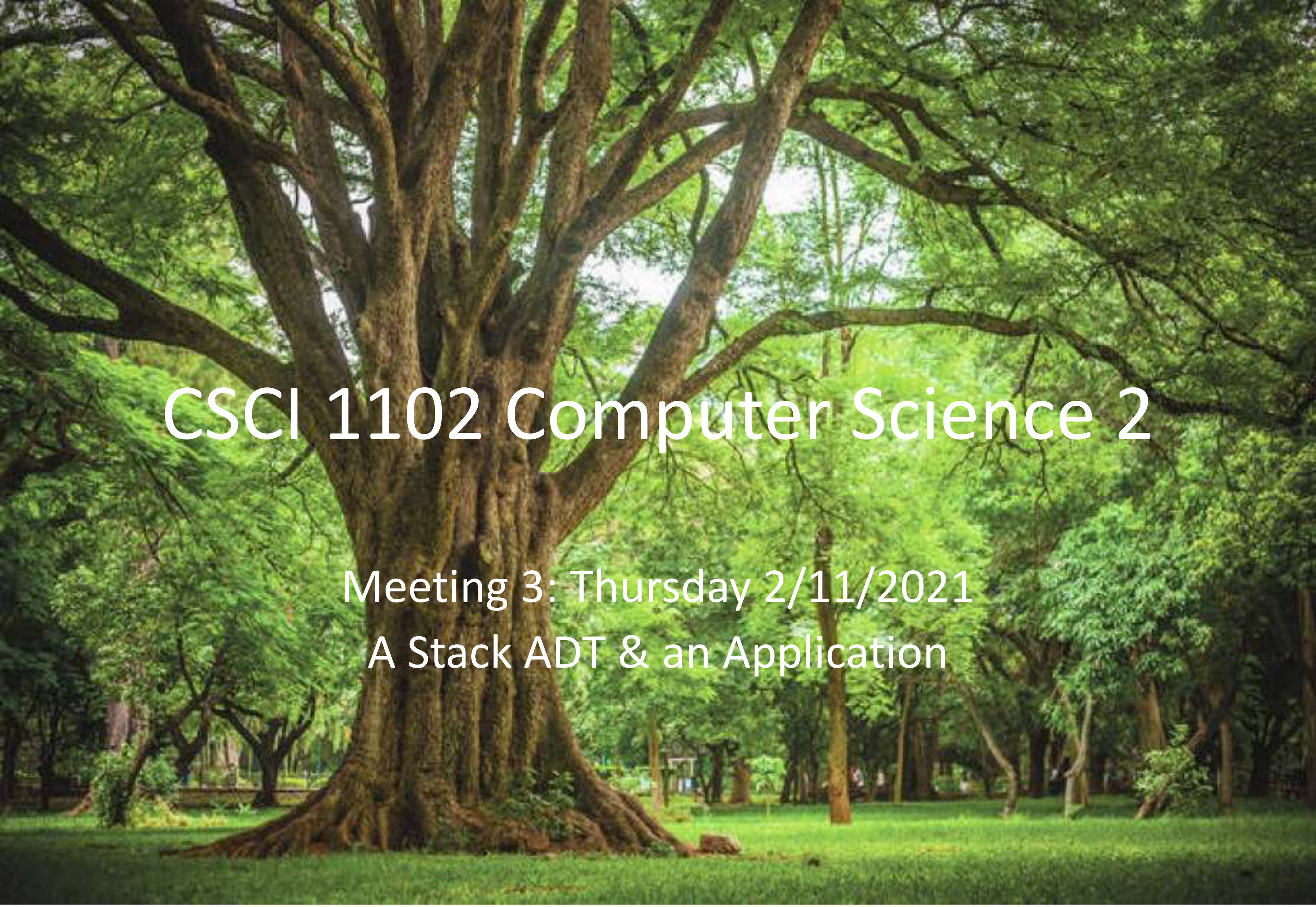


CSCI 1102 Computer Science 2

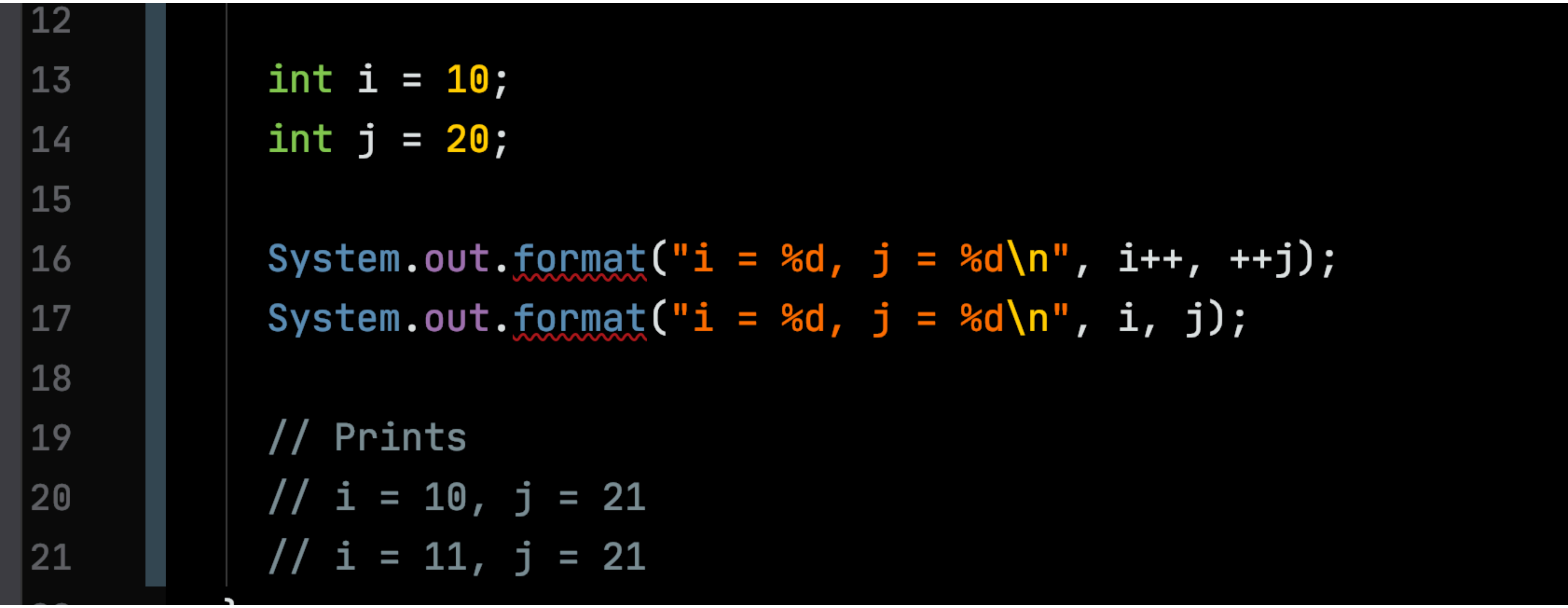
Meeting 3: Thursday 2/11/2021

A Stack ADT & an Application

Today

- A String Stack ADT
- A client: a postfix expression evaluator

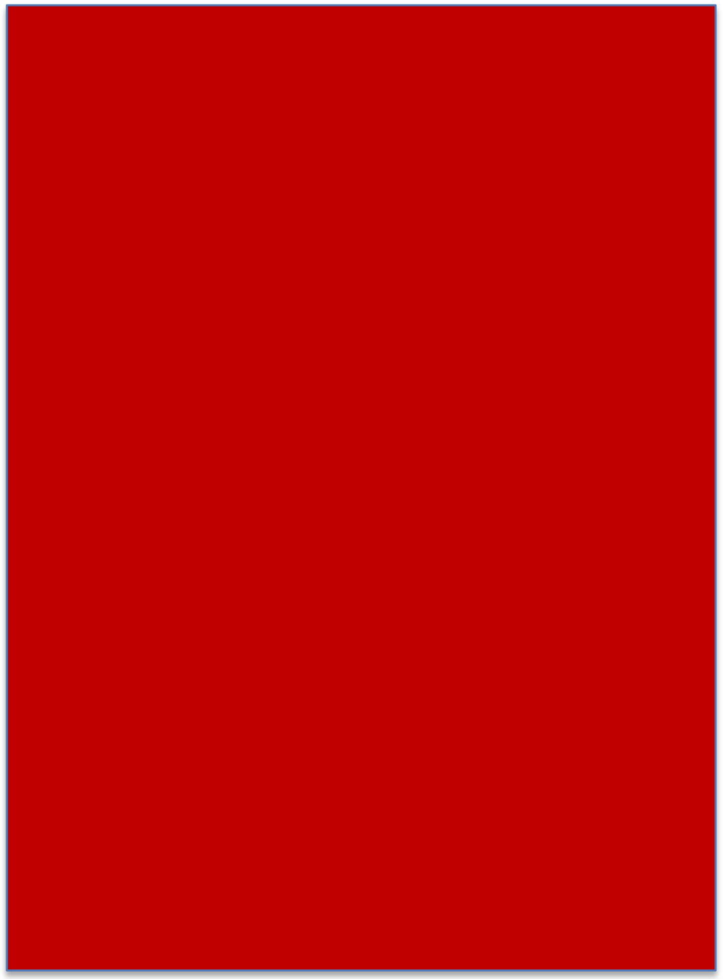
i++ and ++j



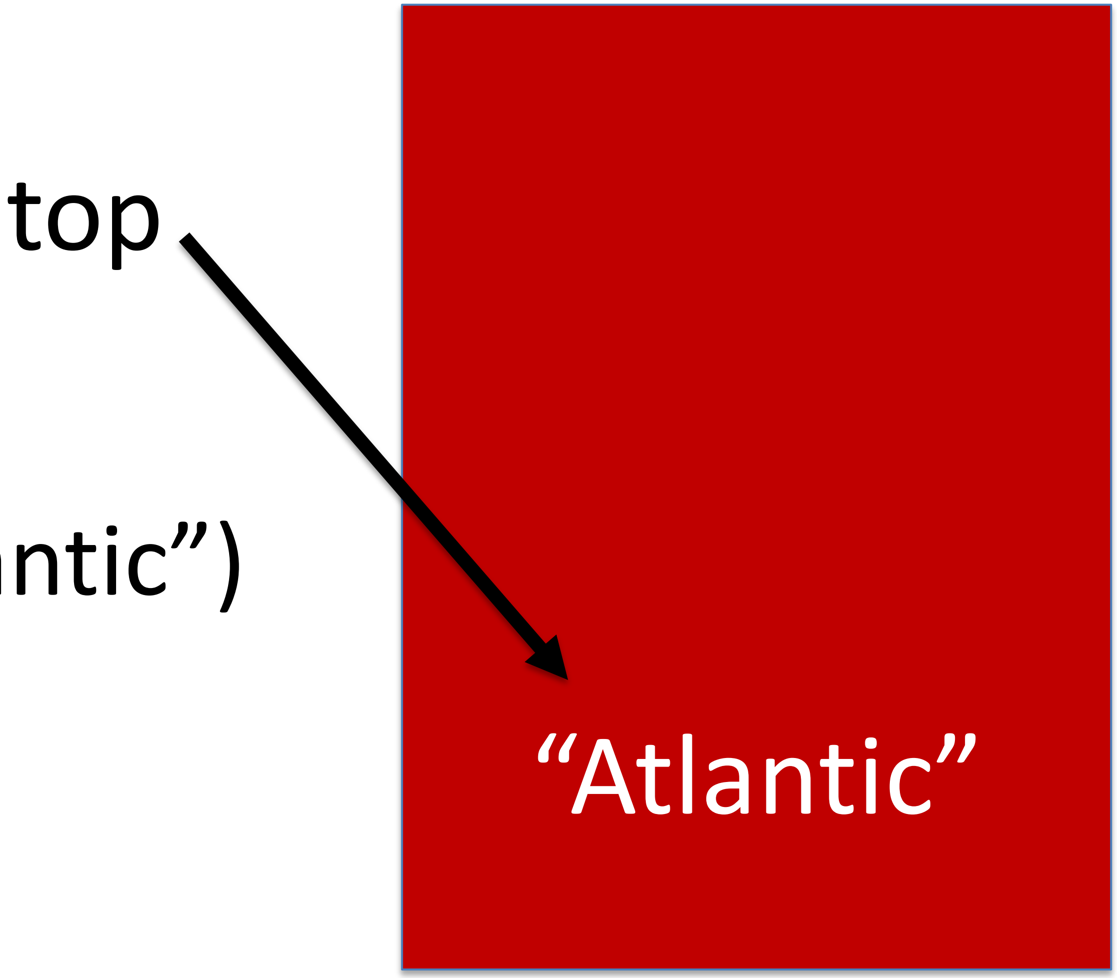
```java
int i = 10;
int j = 20;

System.out.format("i = %d, j = %d\n", i++, ++j);
System.out.format("i = %d, j = %d\n", i, j);

// Prints
// i = 10, j = 21
// i = 11, j = 21
```

A Stack ADT
Last-on, First-off

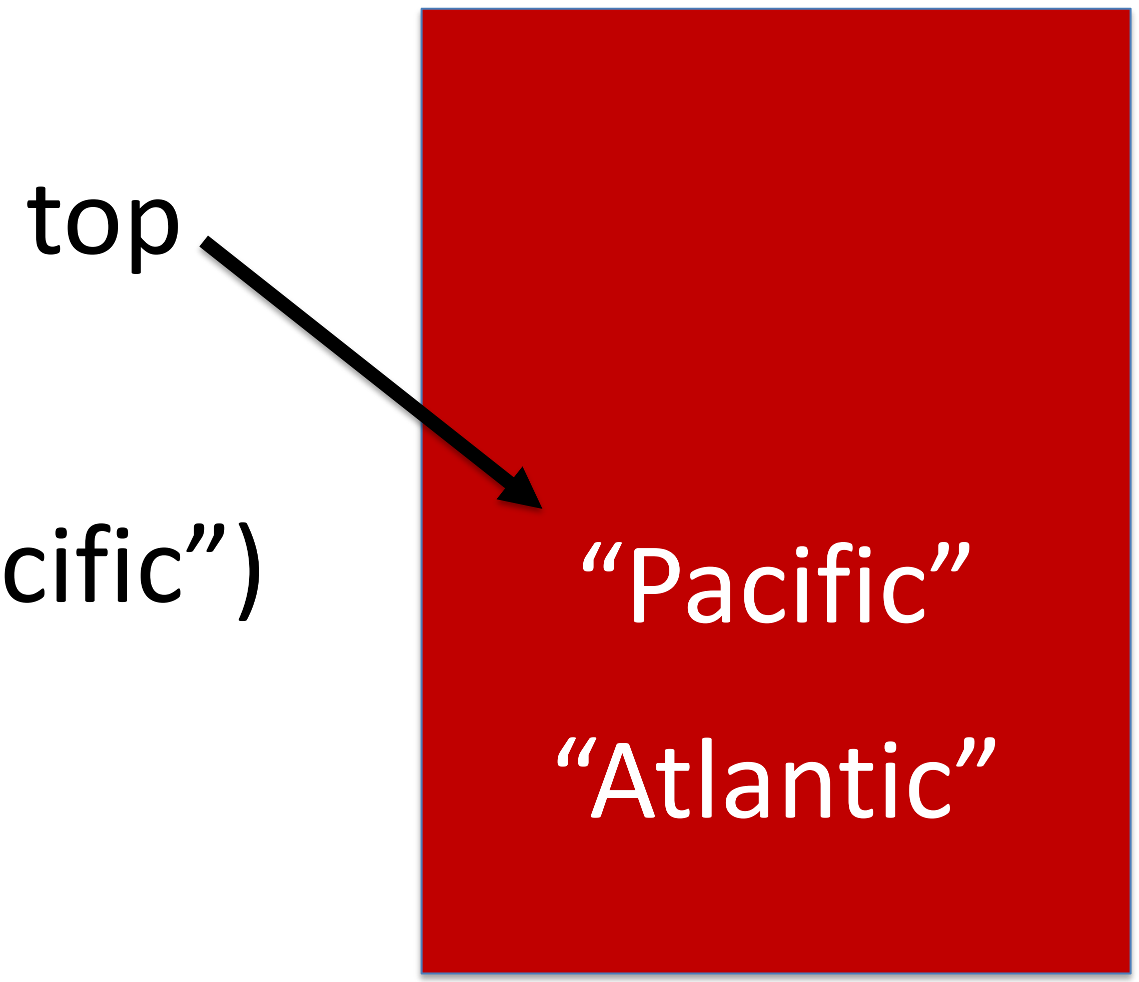
An empty stack

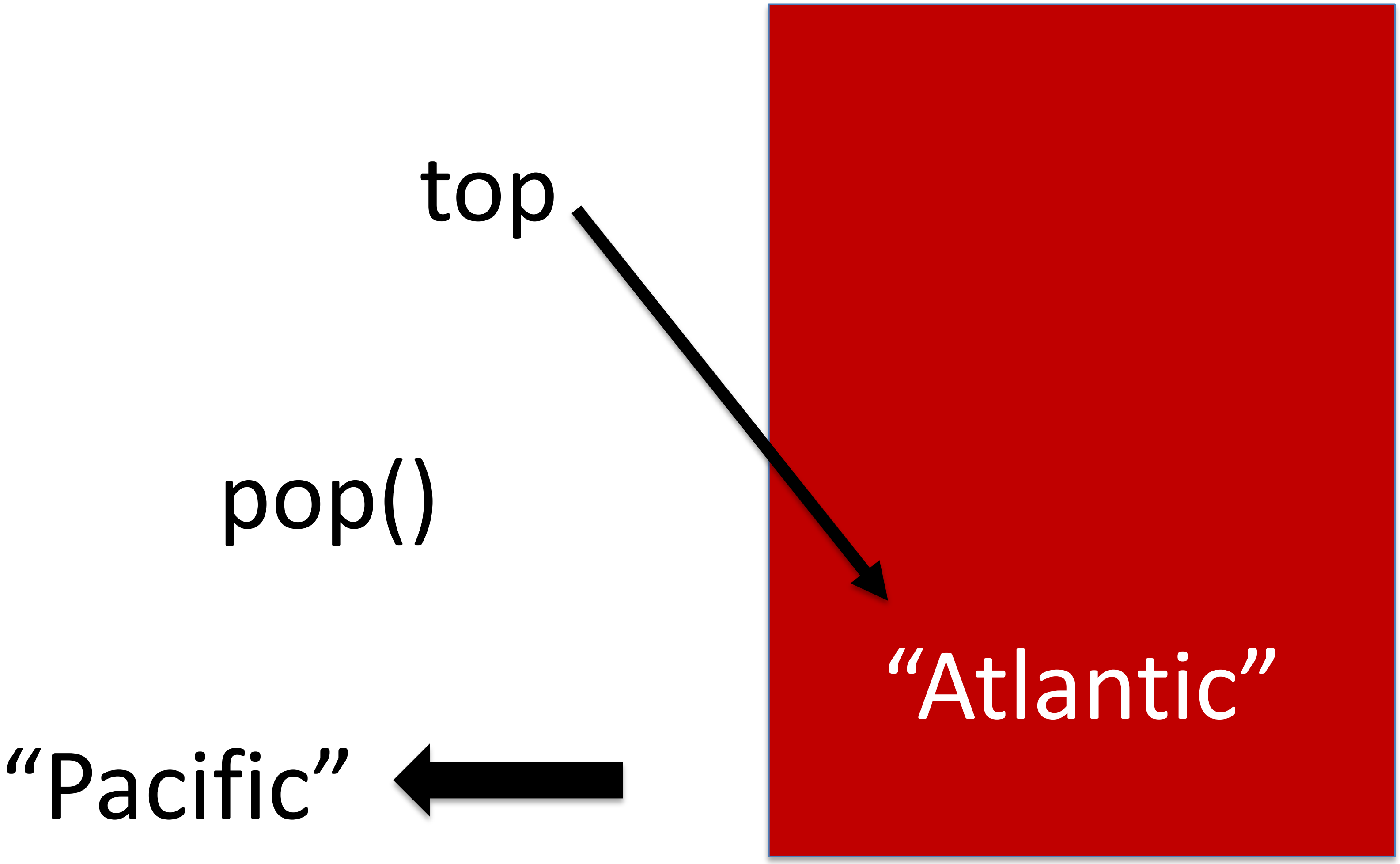


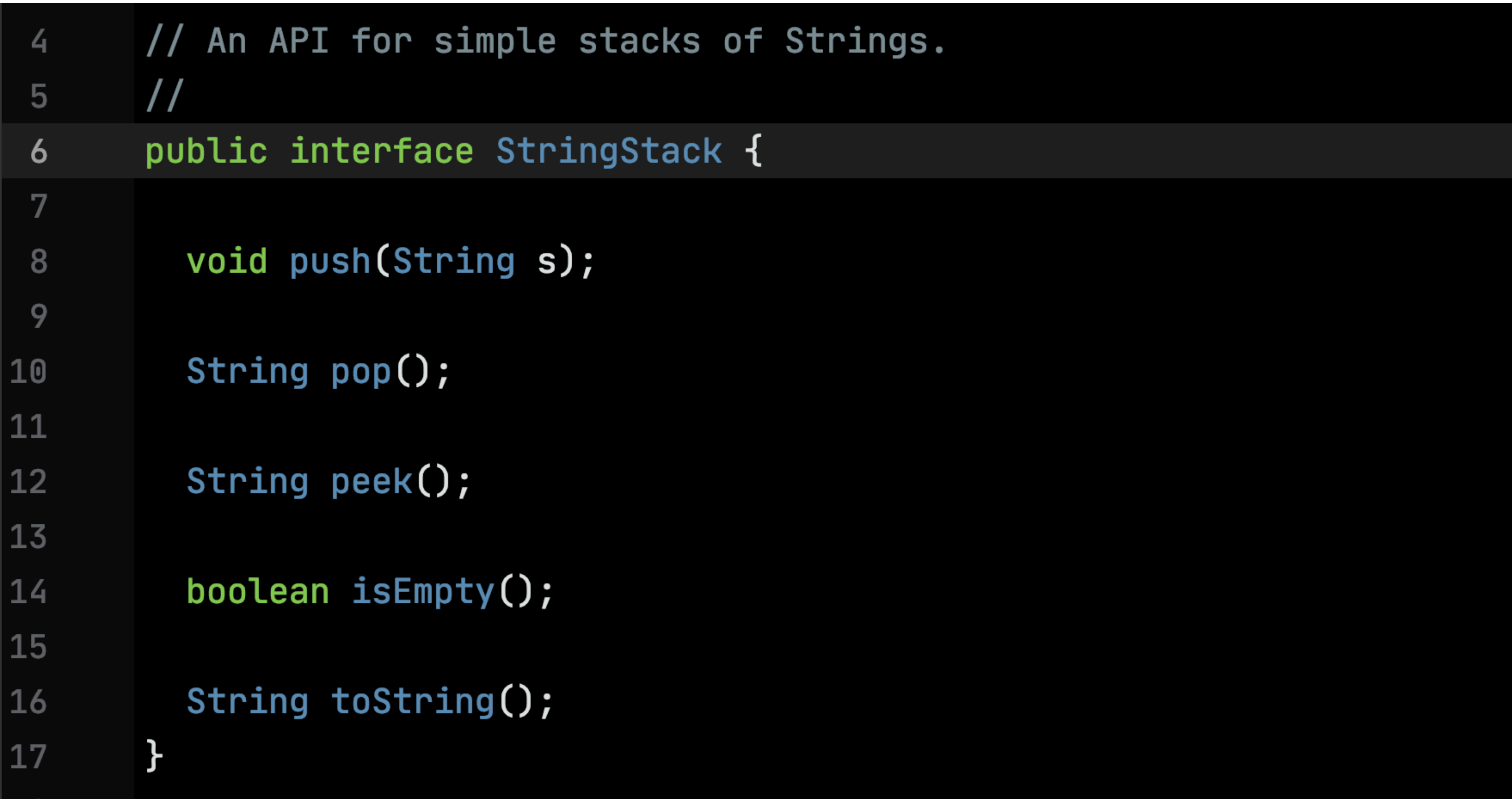
push(“Atlantic”)



push(“Pacific”)




top
pop()
“Atlantic”
“Pacific”


```java
// An API for simple stacks of Strings.
//
public interface StringStack {

  void push(String s);

  String pop();

  String peek();

  boolean isEmpty();

  String toString();
}
```

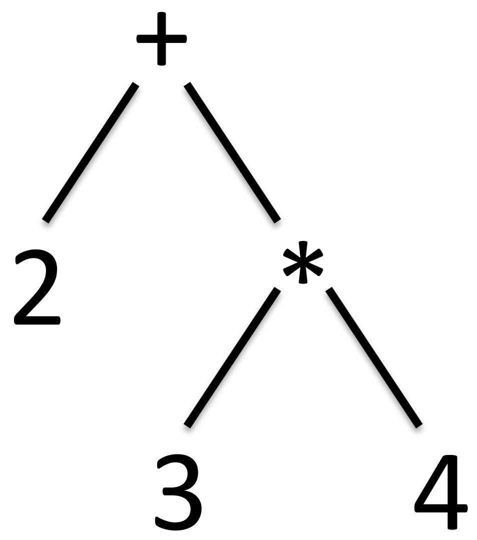
String Stack ADT Code

Postfix Expressions

2 + 3 * 4

$$2 + (3 * 4)$$

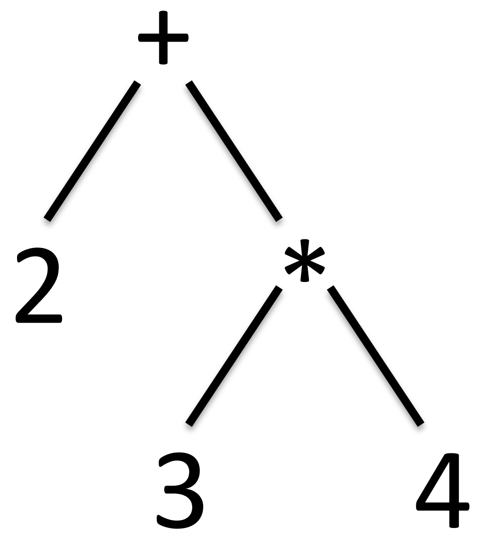
2 + (3 * 4)



Łukasiewicz (Polish/*prefix*) Notation (1924)

Preorder Traversal:

1. Visit the root
2. Traverse the left
3. Traverse the right

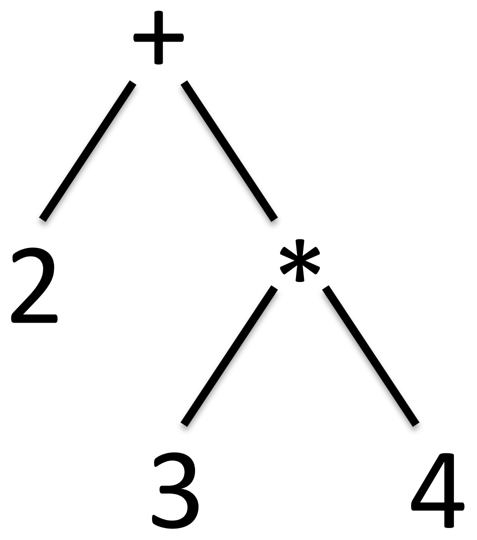


Łukasiewicz (Polish/*prefix*) Notation

Preorder Traversal:

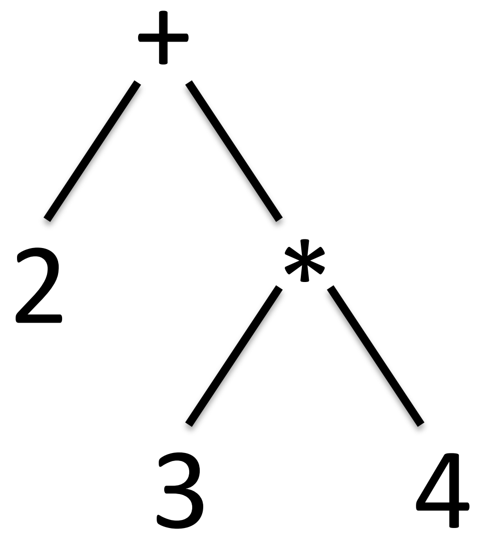
1. Visit the root
2. Traverse the left
3. Traverse the right

+ 2 * 3 4



The Programming Language LISP uses Polish Notation

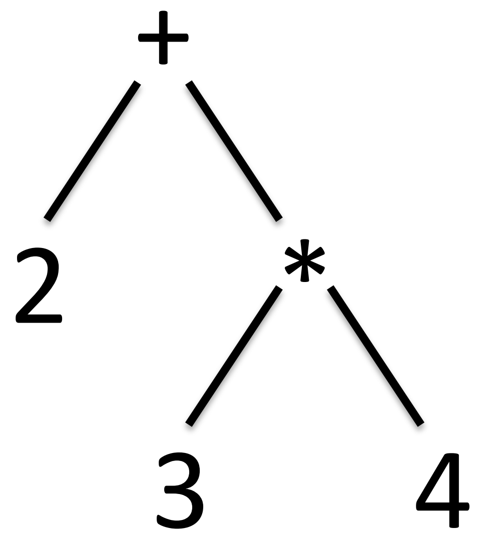
(+ 2 (* 3 4))



Reverse Polish (*postfix*) Notation

Postorder Traversal:

1. Traverse the left
2. Traverse the right
3. Visit the root

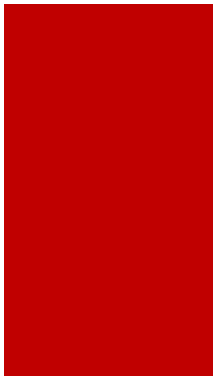


2 3 4 * +

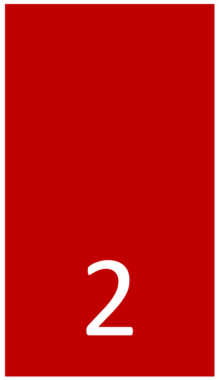
Postfix Notation Evaluation with a Stack

(Burks, Warren & Wright, 1954)

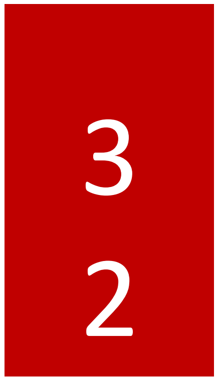
2 3 4 * +



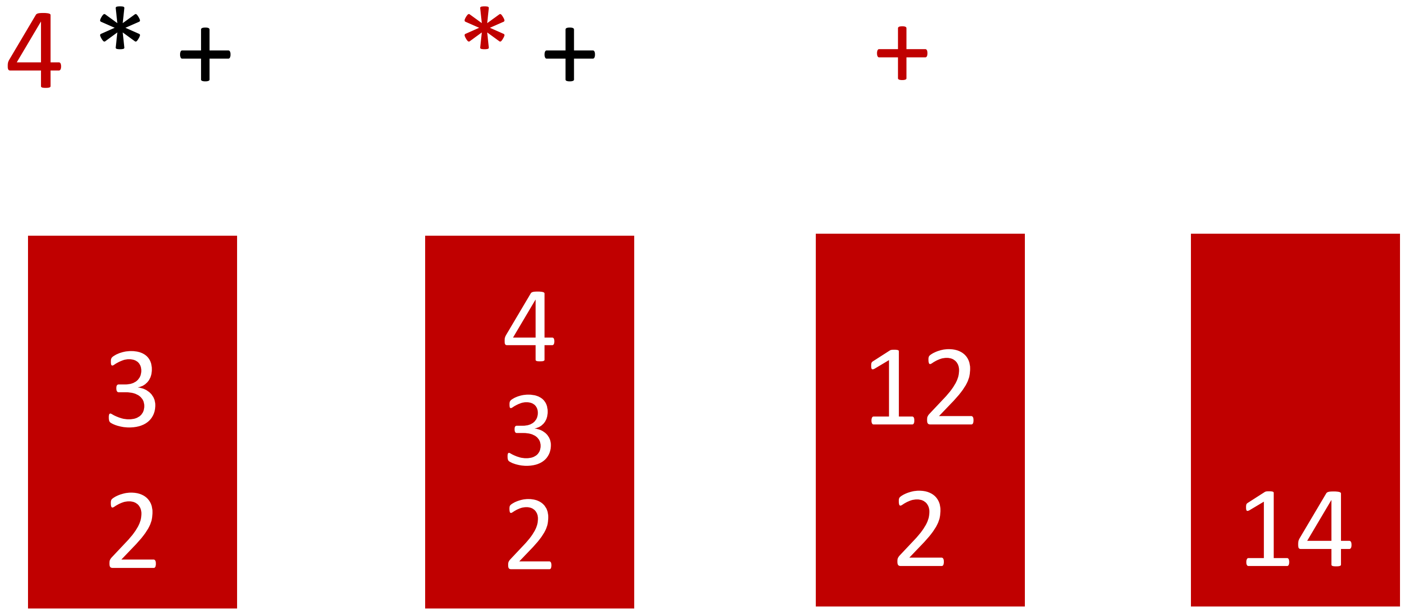
3 4 * +



4 * +



Postfix Notation Evaluation with a Stack



Postfix Evaluation Code