Stacks Performance Task

The prompt:

To read in a group of symbols and check to see if the appropriate opening symbol correctly matches up with the appropriate closing symbol.

The opening symbols are "`{(<[`" and the appropriate closing symbols are "`})>]`".

You must read in and analyze each group.

If you were to read in `{[]}`, you would have a correct balance of opening and closing symbols.
If you were to read in `{[}]`, you would not have a correct balance of opening and closing symbols.

Sample Data :

```
(abc(*def)
[{}]
[
[{<()>}]
{<html[value=4]*(12)>{$x}}
[one]<two>{three}(four)
car(cdr(a)(b)))
car(cdr(a)(b))
```

algorithm hints

When encountering any of the symbols:
only opening symbols should be put on the stack
closing symbols should be compared to the popped top to see if it matches

The process is **true** only after both the expression and stack are empty.

Sample Output :

```
(abc(*def) is incorrect.

[{}] is correct.

[ is incorrect.

[{<()>}] is correct.

{<html[value=4]*(12)>{$x}} is correct.

[one]<two>{three}(four) is correct.

car(cdr(a)(b))) is incorrect.

car(cdr(a)(b)) is correct.
```

The partial class `SyntaxChecker` *is provided to get you started. It already provides a constructor, plus methods to read the symbol groups, convert them to arrays of individual Strings, and other helper methods.*