



CSC 1052 – Algorithms & Data Structures II: Stack Applications

Professor Henry Carter
Spring 2017

Recap

- Stack implementation using a bounded array
- Three phase development
 - ▶ Lather, rinse, repeat
- ADT can be layered to simplify each step
 - ▶ ADT inception

Case Studies

- Break down high-level function
- Implementing the tough bits
- Adversarial behavior



Nested Functions

- Compilers ensure that the functions entered in code are valid
- Often consider paired open and close operations
 - ▶ Parentheses (Lisp)
 - ▶ HTML tags
 - ▶ Case tokens
- Balanced function application



Program Goals

- Input: symbols and an expression
- Balanced or not
- What are the failure cases?



Failure Conditions

- Mismatched symbols
- Imbalanced symbols
- Valid expression



Pseudocode

Tricky Implementation

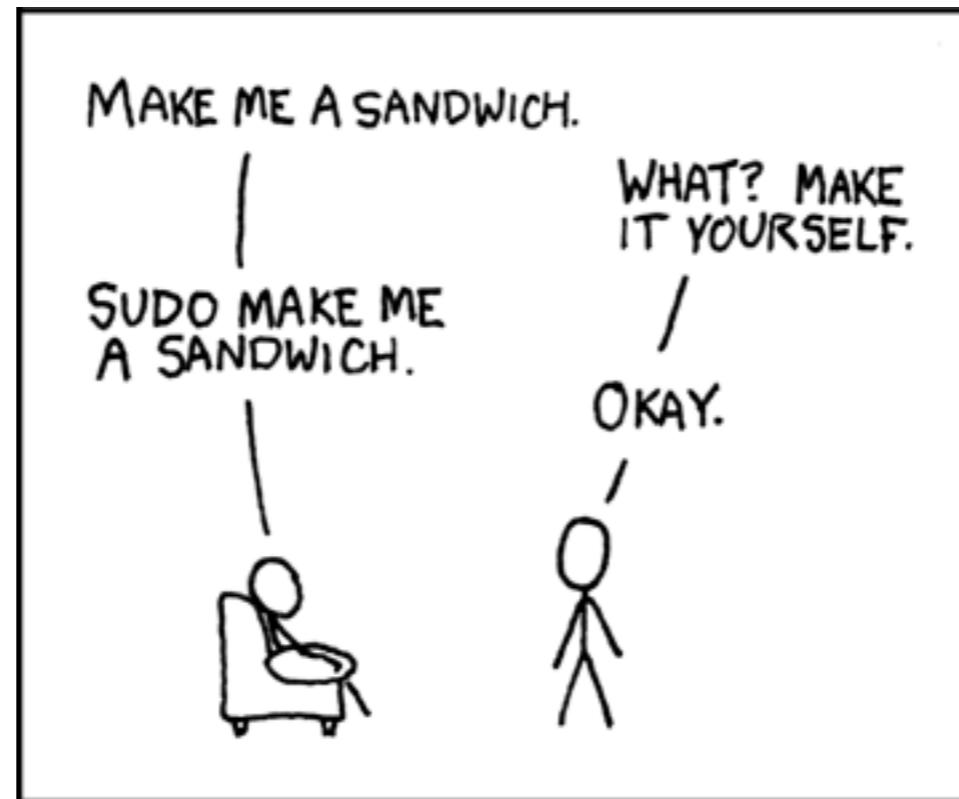
- Symbol comparison
- Loop ordering
- Non-delimiter symbols



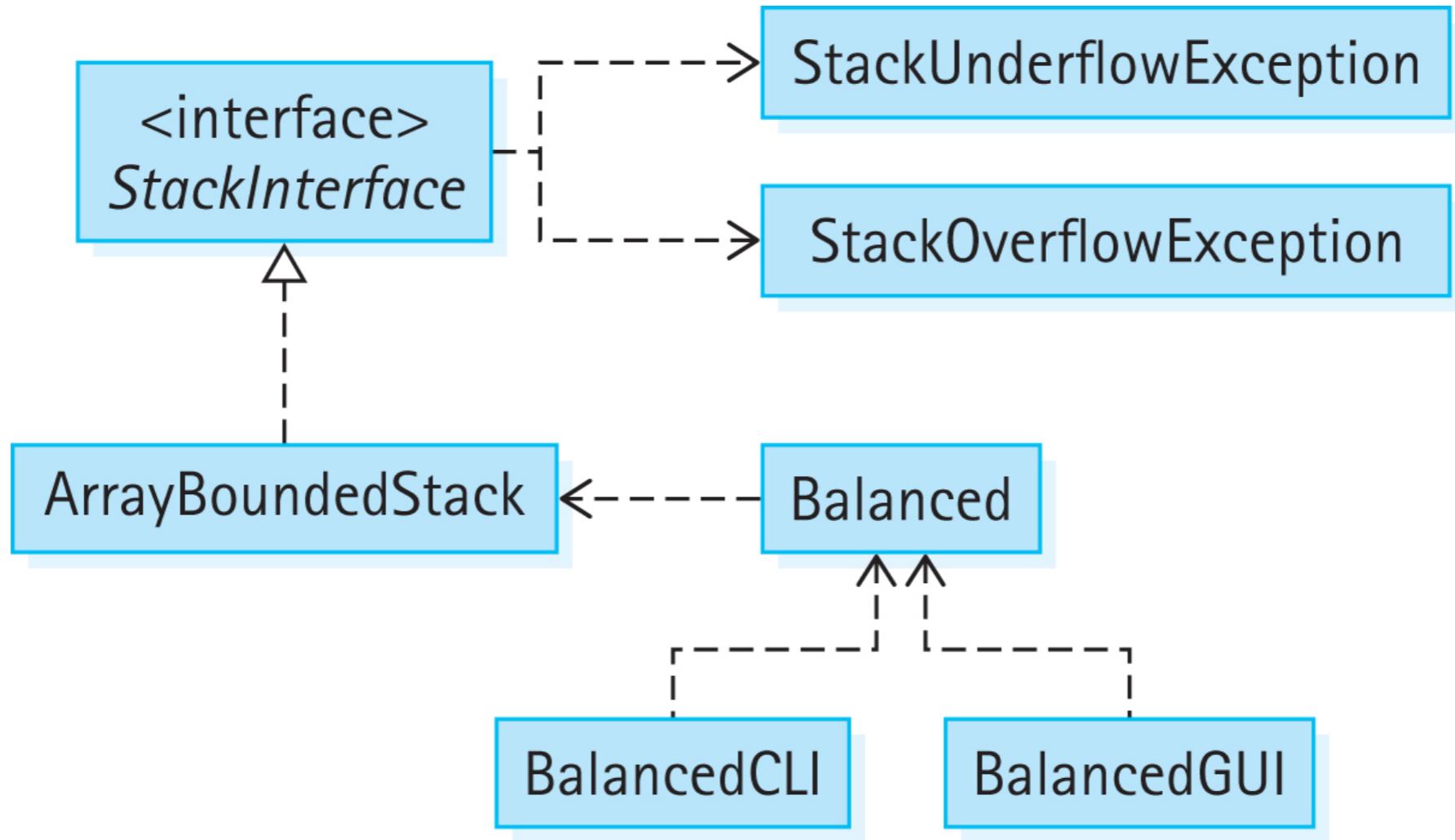
© Yuji Sakai/Ocean/Corbis

Command Line vs GUI

- Ease of development
- Convenience of use
- Most common use-case



Balanced App Architecture



Key:

----> implements
-----> uses

Postfix Expressions

- Typical mathematical equations:
infix expressions
- Requires parentheses,
introduces ambiguity
- Postfix eliminates this complexity



Postfix Expressions

- Infix example: $3 + 4$
- Postfix conversion: $3\ 4\ +$
- Postfix nesting:

$$5 + 6 + (3 - 1) \rightarrow 5\ 6\ 3\ 1\ -\ +\ +$$

More examples

Postfix Expression	Infix Equivalent	Result
4 5 7 2 + - ×	$4 \times (5 - (7 + 2))$	-16
3 4 + 2 × 7 /	$((3 + 4) \times 2)/7$	2
5 7 + 6 2 - ×	$(5 + 7) \times (6 - 2)$	48
4 2 3 5 1 - + × + ×	? × (4 + (2 × (3 + (5 - 1))))	not enough operands
4 2 + 3 5 1 - × +	$(4 + 2) + (3 \times (5 - 1))$	18
5 3 7 9 + +	$(3 + (7 + 9)) \dots 5???$	too many operands

Program Goals

- Input: symbols, integers, and an expression
- Output: evaluation or error
- Error conditions?



MoHPC

Error Conditions

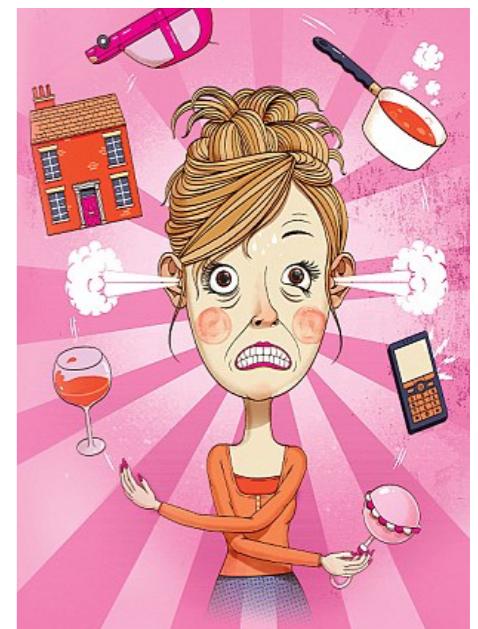
- Too many operands
- Not enough operands
- Illegal symbols

02654895465	23421404359	85123030213	02654895465	23421404359
13025165465	78553402213	13311000011	13025165465	78553402213
76540215497	49758672464	25468952654	76540215497	49758672464
87654860216	97968652031	78021328503	87654860216	97968652031
54897564202	25679561203	57920045685	54897564202	25679561203
15465465460	26456530979	48314904153	15465465460	26456530979
21654	21503	0811001240	16346316748	21654621503
40216	40216	0811001240	16346316748	2123
56102	56102	0811001240	16346316748	4545
62165	62165	0811001240	16346316748	5425
13245450154	34659782135	35656497652	13245450154	34659782135
84987984301	64023100002	31200124556	84987984301	64023100002
24568765435	13656462857	87976423120	24568765435	13656462857
01235435435	55645622256	31655976421	01235435435	55645622256
43021648576	79866566433	05234605242	43021648576	79866566433
53441100000	59823101346	59257561221	53441100000	59823101346
00000001243	56457242104	56024565237	00000001243	56457242104
53727672034	23168976543	85421245454	53727672034	23168976543
25375763520	24212124567	45456402124	25375763520	24212124567
43597572672	54212054976	24575454012	43597572672	54212054976
40133727967	85323051564	42245454440	40133727967	85323051564
97801322479	65246791630	55546520303	97801322479	65246791630
69675014372	21352675642	40555120245	69675014372	21352675642

Pseudocode

Tricky Implementation

- Reading single tokens (input must be space-separated)
- Identifying when to throw exceptions
- Remember to top AND pop



Recap

- Balanced expressions ensures delimiters are matched using a stack
- Postfix evaluation reduces math expression ambiguity at the cost of readability
- Remember:
 - ▶ Code
 - ▶ Test
 - ▶ Repeat

Next Time...

- Dale, Joyce, Weems Chapter 2.7
 - ▶ Remember, you need to read it BEFORE you come to class!
- Check the course webpage for practice problems
- Peer Tutors
 - ▶ <http://www.csc.villanova.edu/help/>

