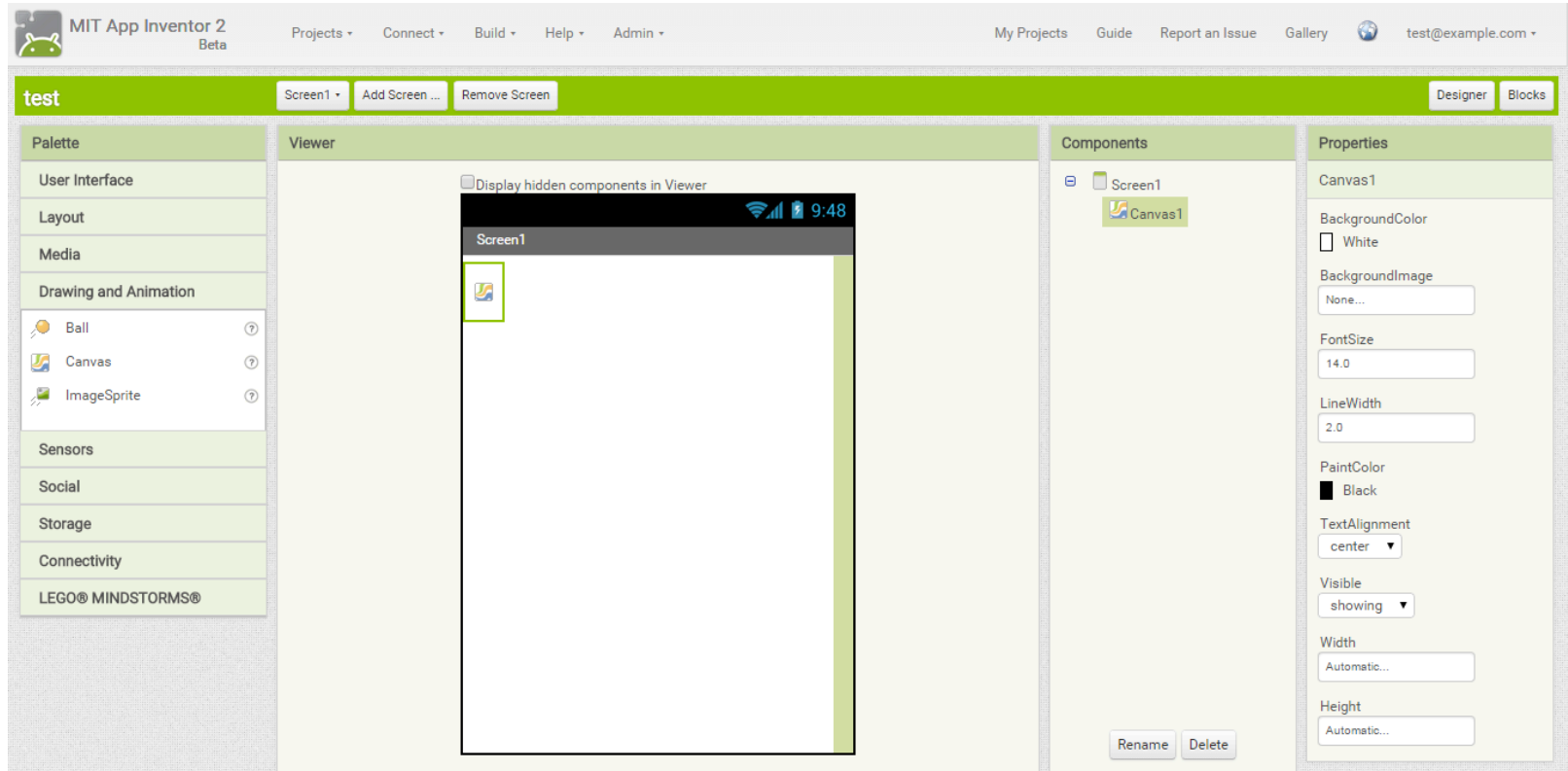# Adapting higher-order list operators for blocks programming

Soojin Kim and Franklyn Turbak

Wellesley College

October 21, 2015

Designer

Blocks Editor

Problems with Current List Operators

Current List Operators

create empty list

make a list

add items to list   list
item

is in list? thing
list

length of list   list

is list empty?   list

pick a random item   list

index in list thing
list

select list item   list
index

insert list item list
index
item

replace list item   list
index
replacement

remove list item   list
index

append to list   list1
list2

copy list   list

is a list?   thing

list to csv row   list

list to csv table   list

list from csv row   text

list from csv table   text

look up in pairs key
pairs
notFound

Berry's Lemonade

$2

Date	# of Lemonades Sold	Daily Profit ($)
6/1/15	13	26
6/2/15	20	40
6/3/15	N/A	
6/4/15	18	36
6/5/15	N/A	
6/6/15	10	20
6/7/15	16	32
	Total Profit	154

Berry's Lemonade List Data in App Inventor

Solution Using a Loop and Local Variable
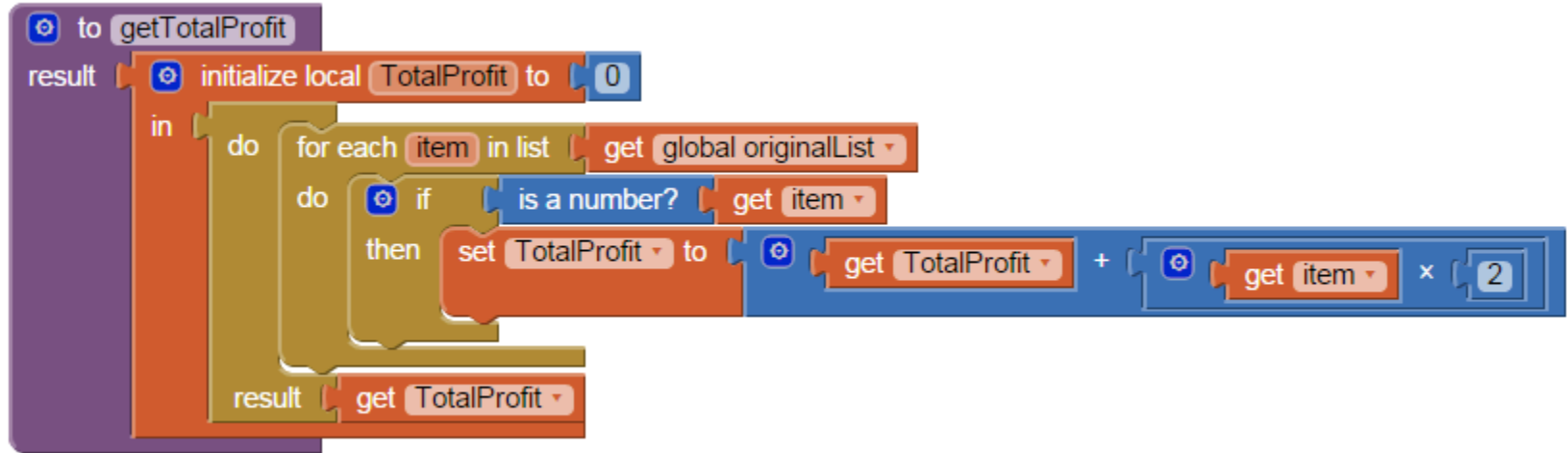
Solution Using a Loop and Global Variable

Common Bug: Failure to Reinitalize Global Loop Variable

Sort: Old Design
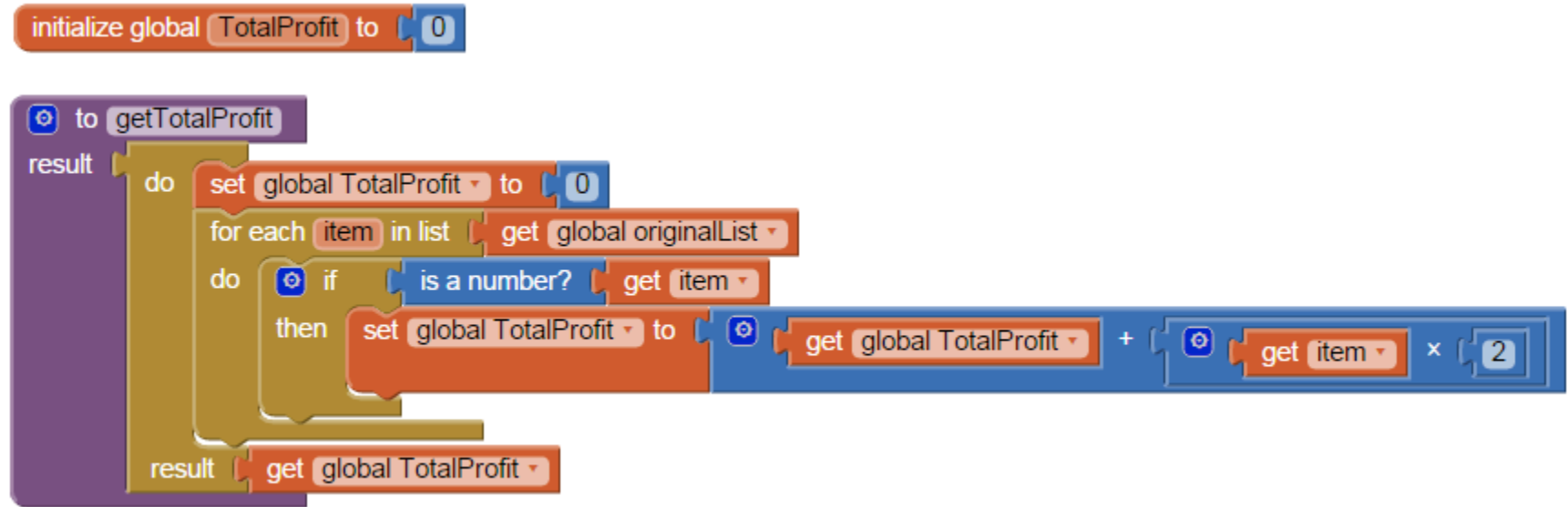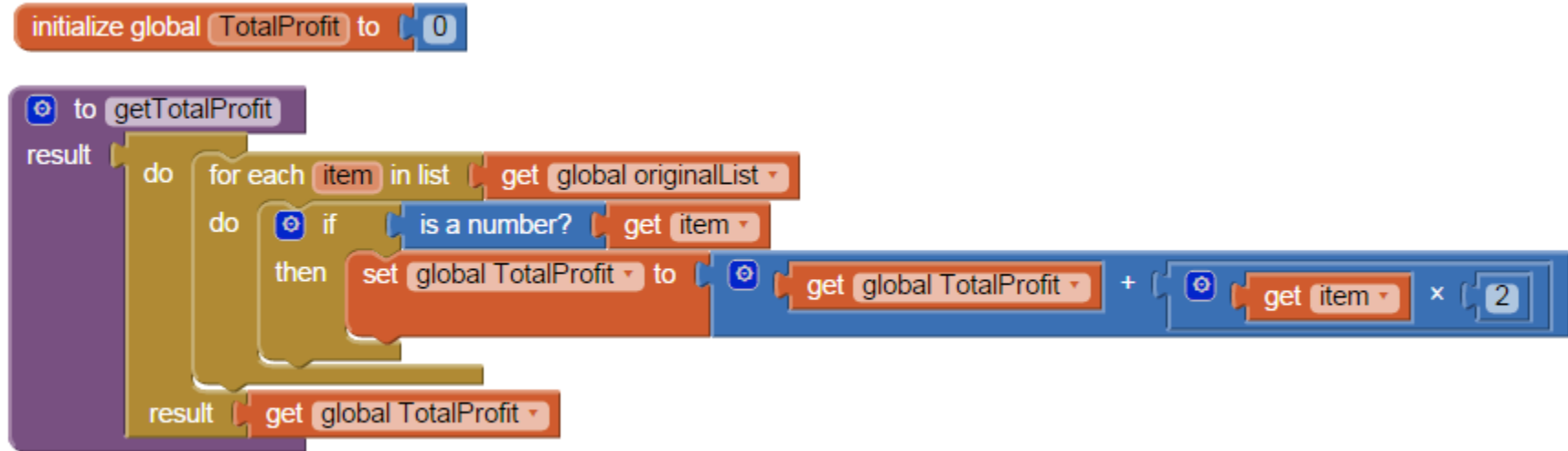
http://www.imagnity.com/tutorials/app-inventor/list-sorting-on-app-inventor/
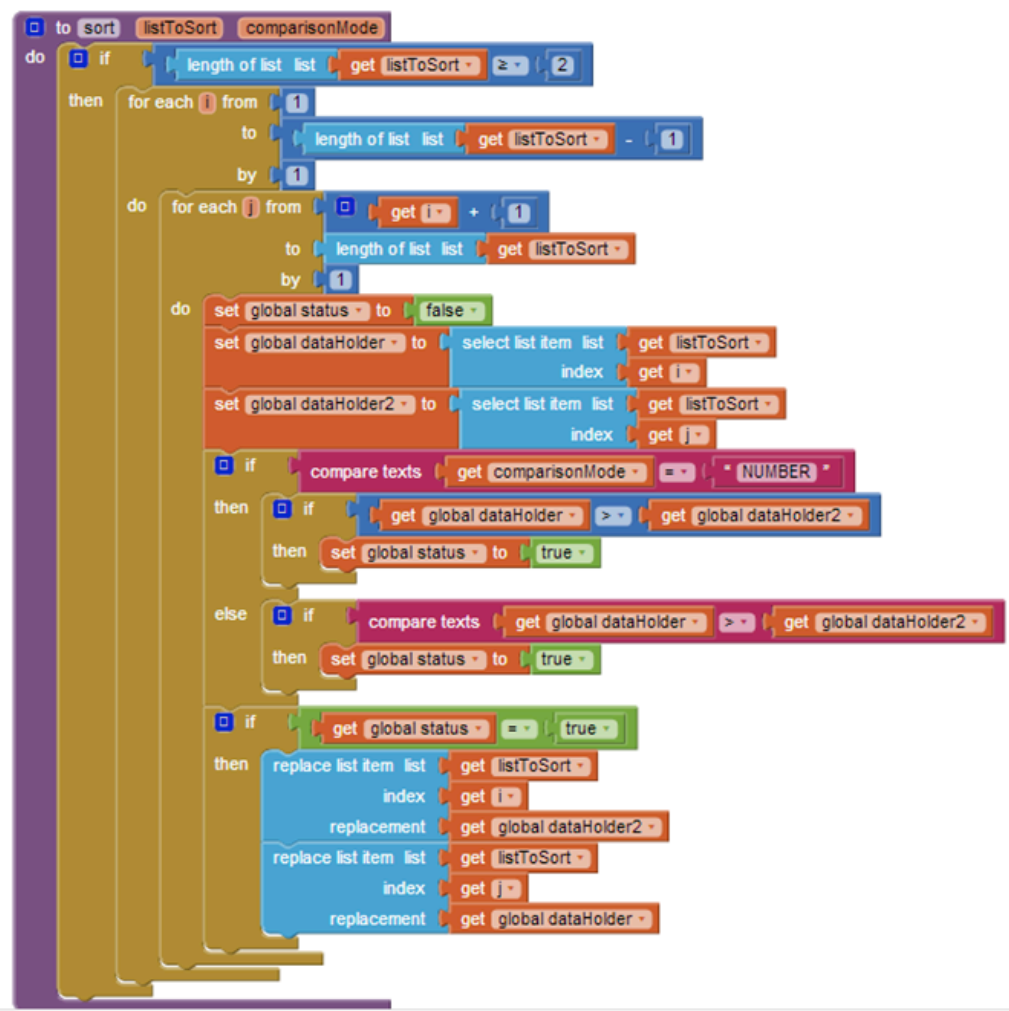
Solution: Addition of Pseudo-Higher-Order Operators (PHOLOs)

Map, Filter, and Reduce in Python

>>>(map (lambda x: x + 1) [5,3,8,10,2])
 [6,4,9,11,3]

>>>(filter (lambda x: x < 6) [5,3,8,10,2])
[5,3,2]

>>>(reduce (lambda x, y: x + y) [5,3,8,10,2])
28

Higher-Order List Operators (HOLOs)

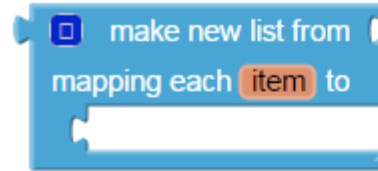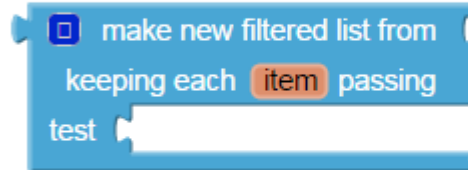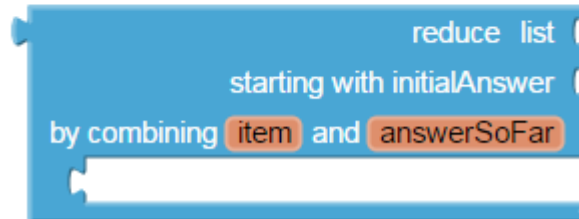# Map, Filter, and Reduce in App Inventor

Map

Filter

Reduce

Map Block

$$(1) \quad (2) \quad (3)$$
map (lambda item: item + 1, [5,3,8,10,2])
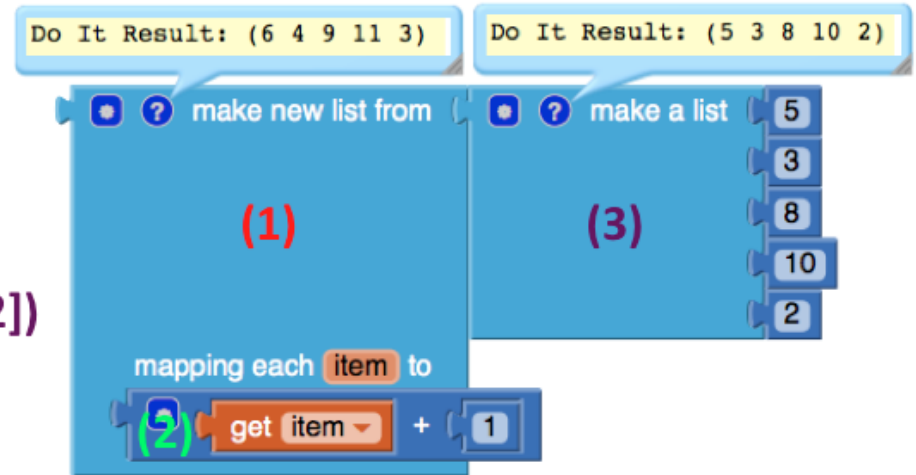
Pseudo-Higher-Order Operators (PHOLOs)

Solution Using Filter, Map, and Reduce
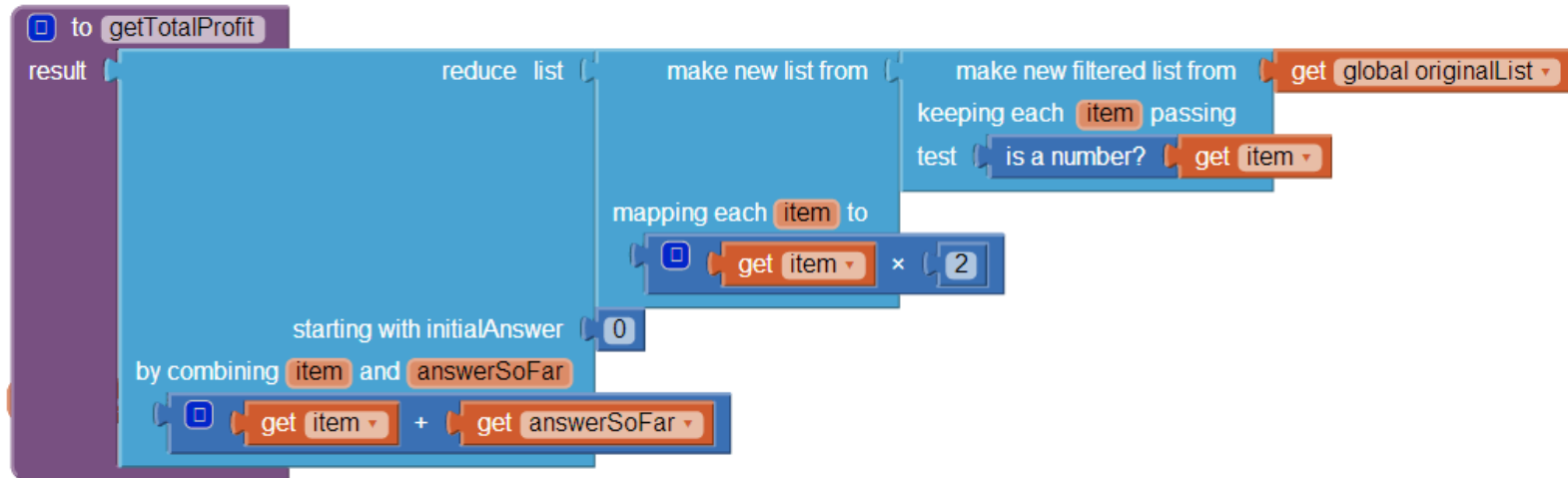
Three Sort Blocks

Basic sort

Sort with key
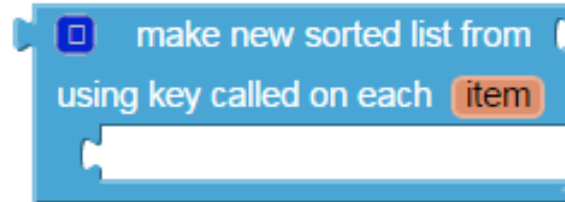
Sort with comparator

Basic Sort

Sort with Key

Do It Result: (Zoe Sam Alex David Megan Brendan)

Do It Result: (David Brendan Zoe Sam Alex Megan)
---

make new sorted list from   get global names   ⇦ input list
using key called on each item
length get item ⇦ key expression

Sort with Comparator

Do It Result: (Zoe Sam Megan David Brendan Alex)

Do It Result: (David Brendan Zoe Sam Alex Megan)
---

make new sorted list from   get global names   ⇐ input list

where item1 precedes item2 if
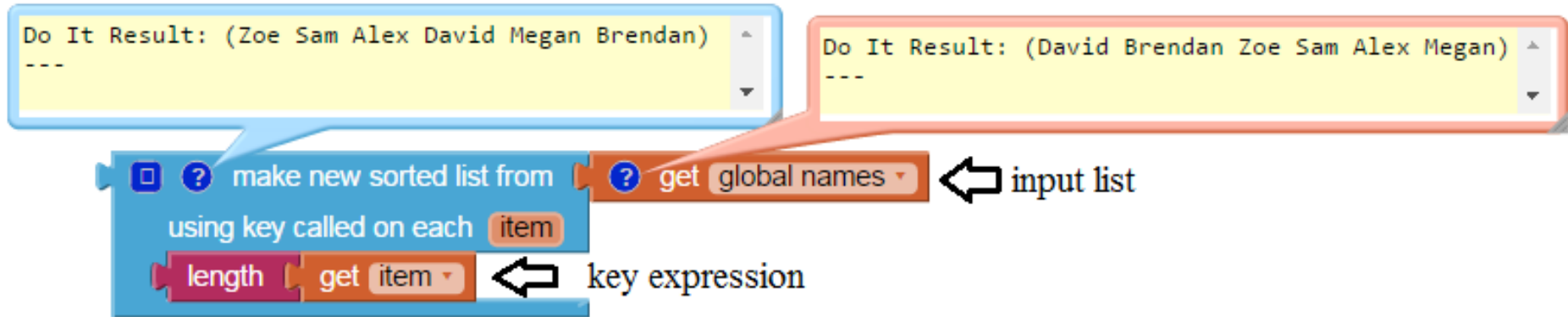
compare texts   get item1   > ▾   get item2   ⇐ comparator

Design and Results of User Study

Design of User Study

- 60 to 90 minutes

- Short tutorial on each list operator

- Part 1: 8 tasks involving mapping, filtering, and/or reducing

- Part 2: 6 tasks involving sorting

User Study Participants

- 18 Wellesley students who had previous experience working with App Inventor

- 10 users (56%) knew map, filter or reduce previously and the remaining 8 users (44%) did not

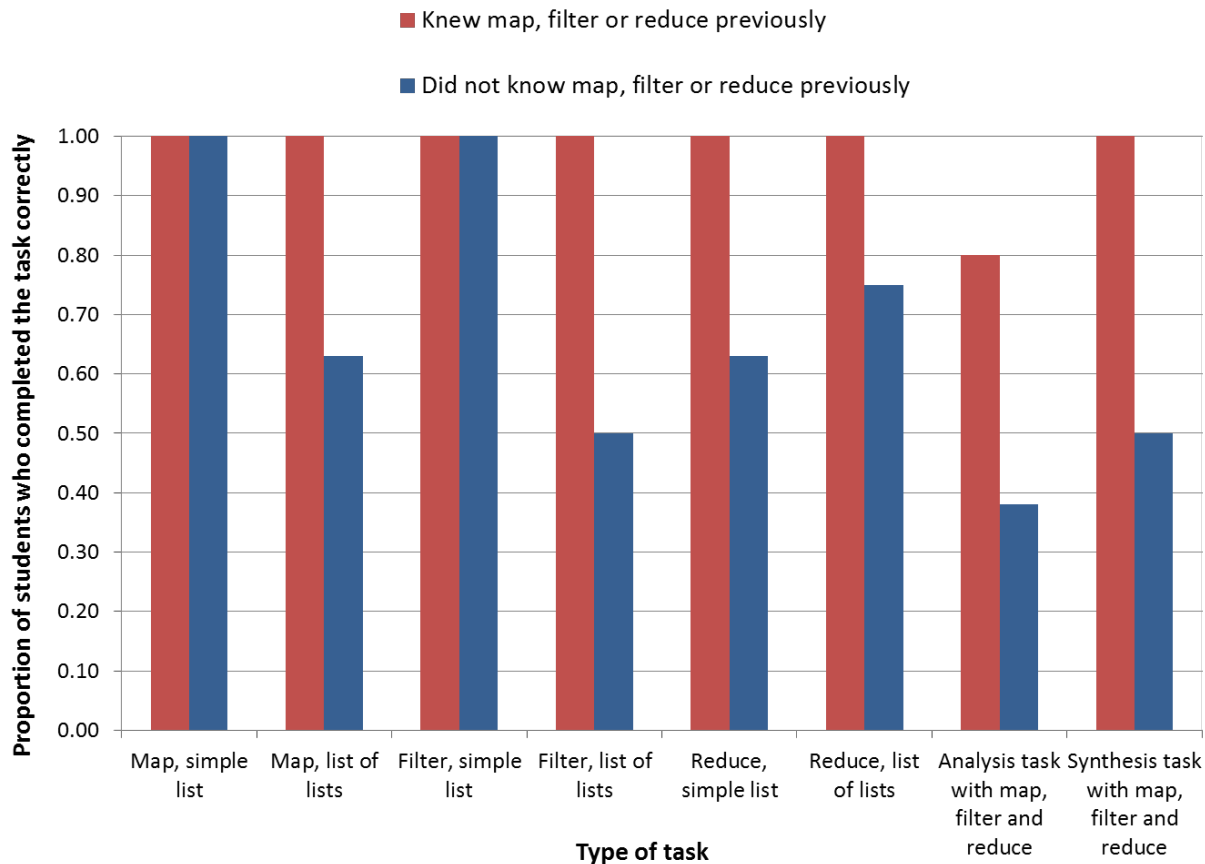# User Study Results Part 1

Feedback

- "I have worked with map, filter, and reduce a lot in different languages, so the concepts were familiar and I was able to interpret pretty quickly what parts the blocks should have. "

- "These blocks were fairly simple to use, but I sometimes became frustrated because I would forget which block was useful in what kind of scenario. Reading the english on the blocks also helped with this though when I would get stuck."

User Study Results Part 2
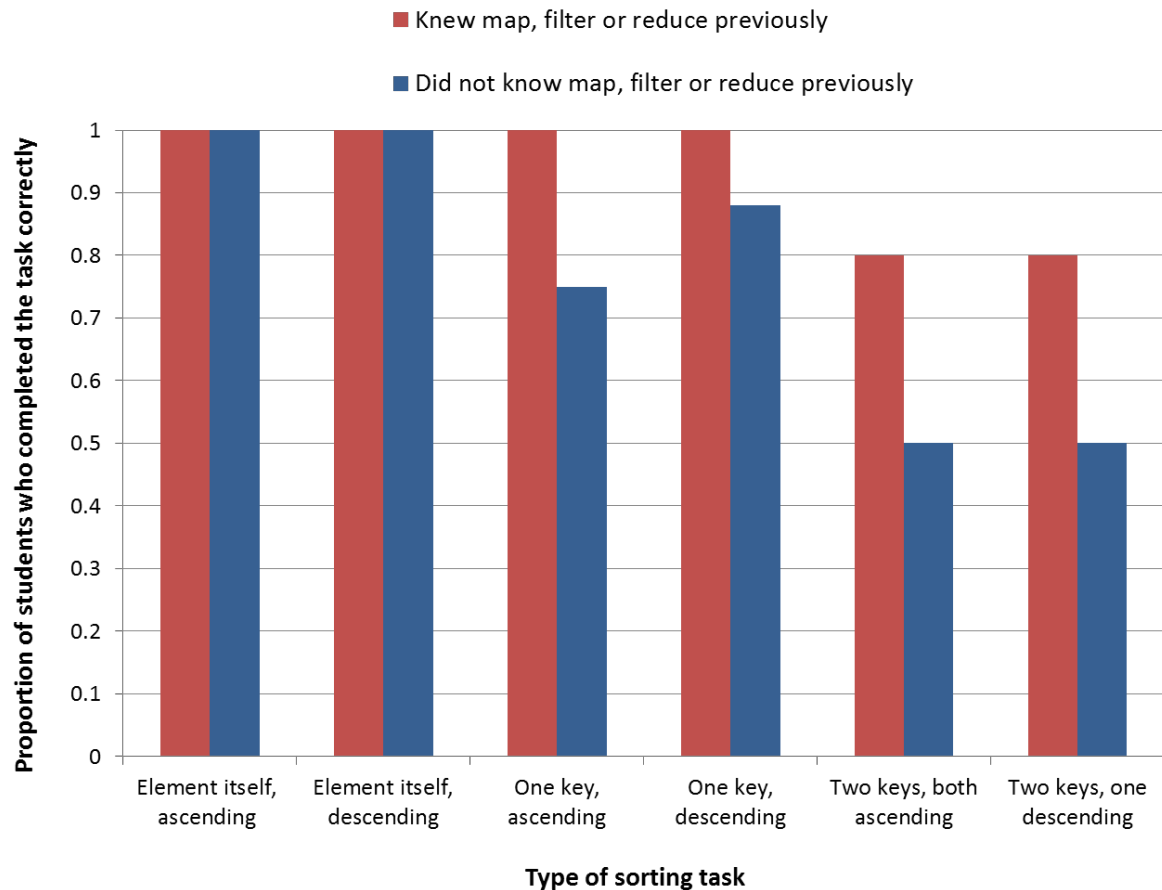
Feedback

- "I didn't like that there were three different blocks for three different kinds of sorting...I almost think it would be easier if you had to explicitly decide how you want a list to be sorted every time you want to sort a list."

- "I liked that there were three options, so I could use the one I felt most comfortable with."

- "There are multiple ways you can perform a single task, especially with these three specific sort blocks. That made it both easier (can use any) and more difficult (many options for how to execute) to use."

Related Work

Map in Snap! vs. App Inventor

Canvas App Example in Code.org App Lab


```
1   button("clearButton", "clear");
2   createCanvas("myCanvas", 200, 300);
3   setActiveCanvas(▼"myCanvas");
4   onEvent(▼"myCanvas", ▼"mousedown", function(event) {
5       setFillColor(▼"blue");
6       circle(event.clientX, event.clientY, 10);
7   });
8   onEvent(▼"clearButton", ▼"click", function(event) {
9       clearCanvas();
10  });
```

Properties of Events in Code.org App Lab vs. App Inventor

Conclusions and Future Work

Conclusions

- Addition of PHOLO blocks
  - Understandable to some extent
  - Bypasses problem of not having first class functions

- Sort with key not necessary

Future Work

- Finalize the labels on the blocks

- Conduct additional user studies
  - Compare loops vs. PHOLOs
  - Study wider demographic base

- In the process of adding to official App Inventor release

Questions?