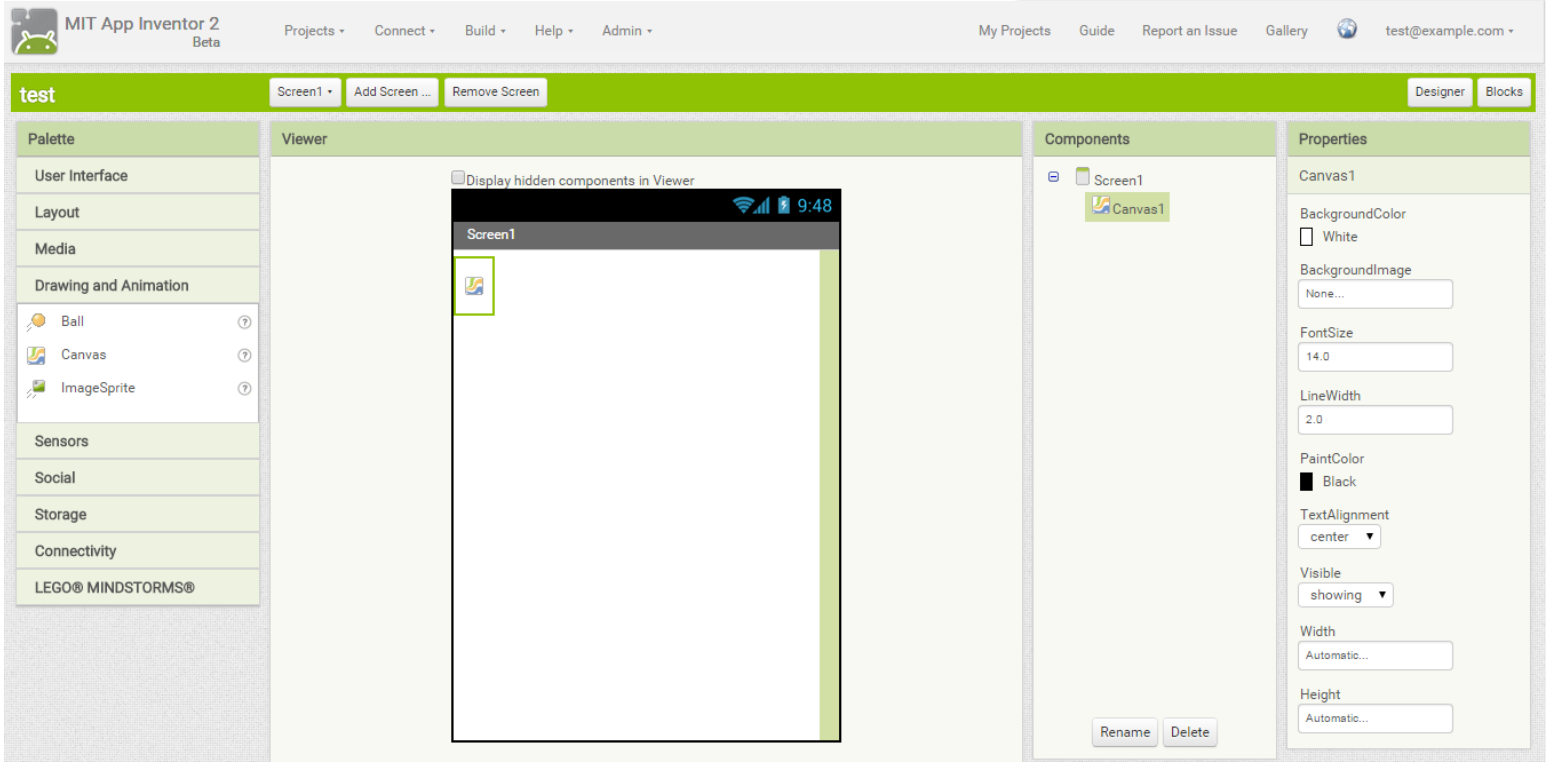
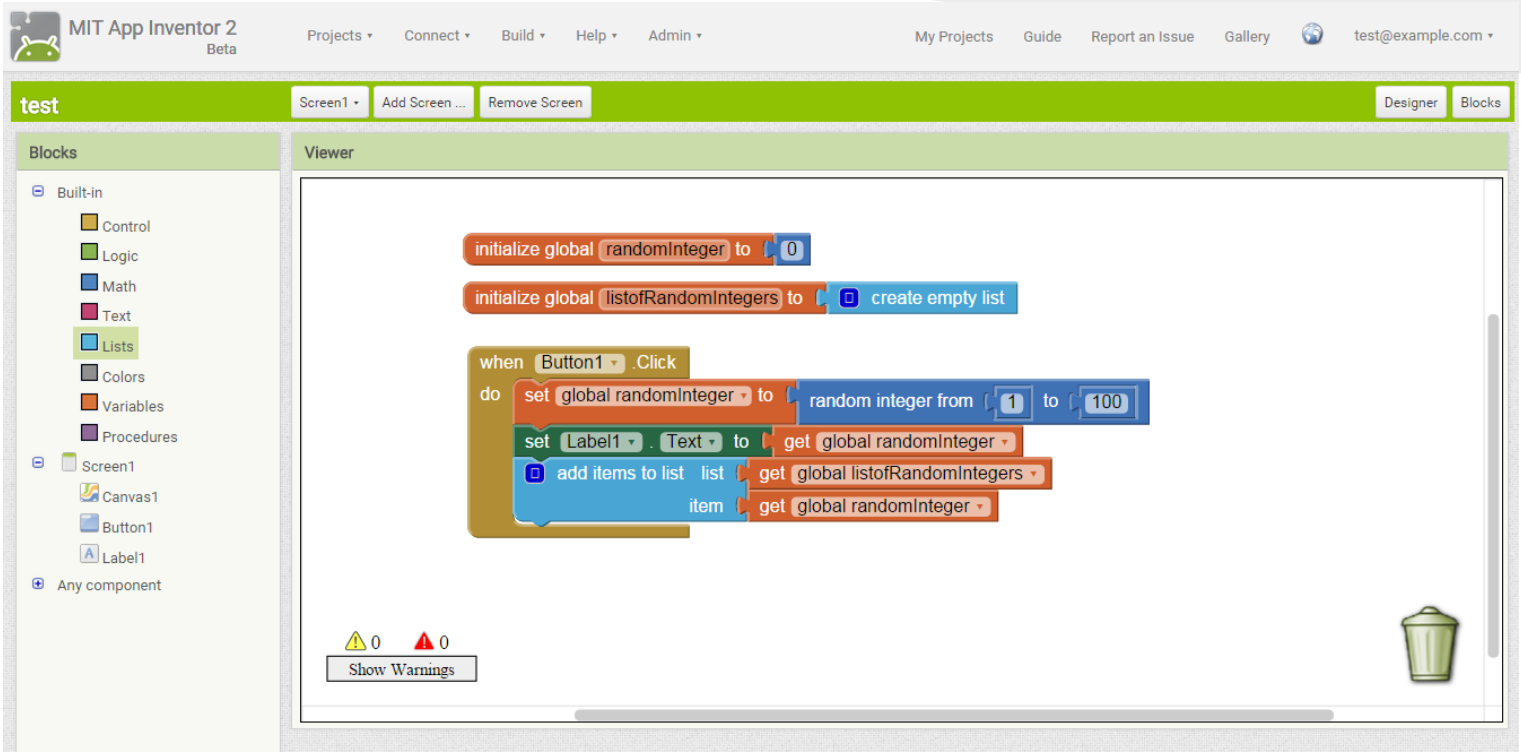
Developing and Assessing New List Operators in App Inventor

Soojin Kim
Advisor: Franklyn Turbak

Designer

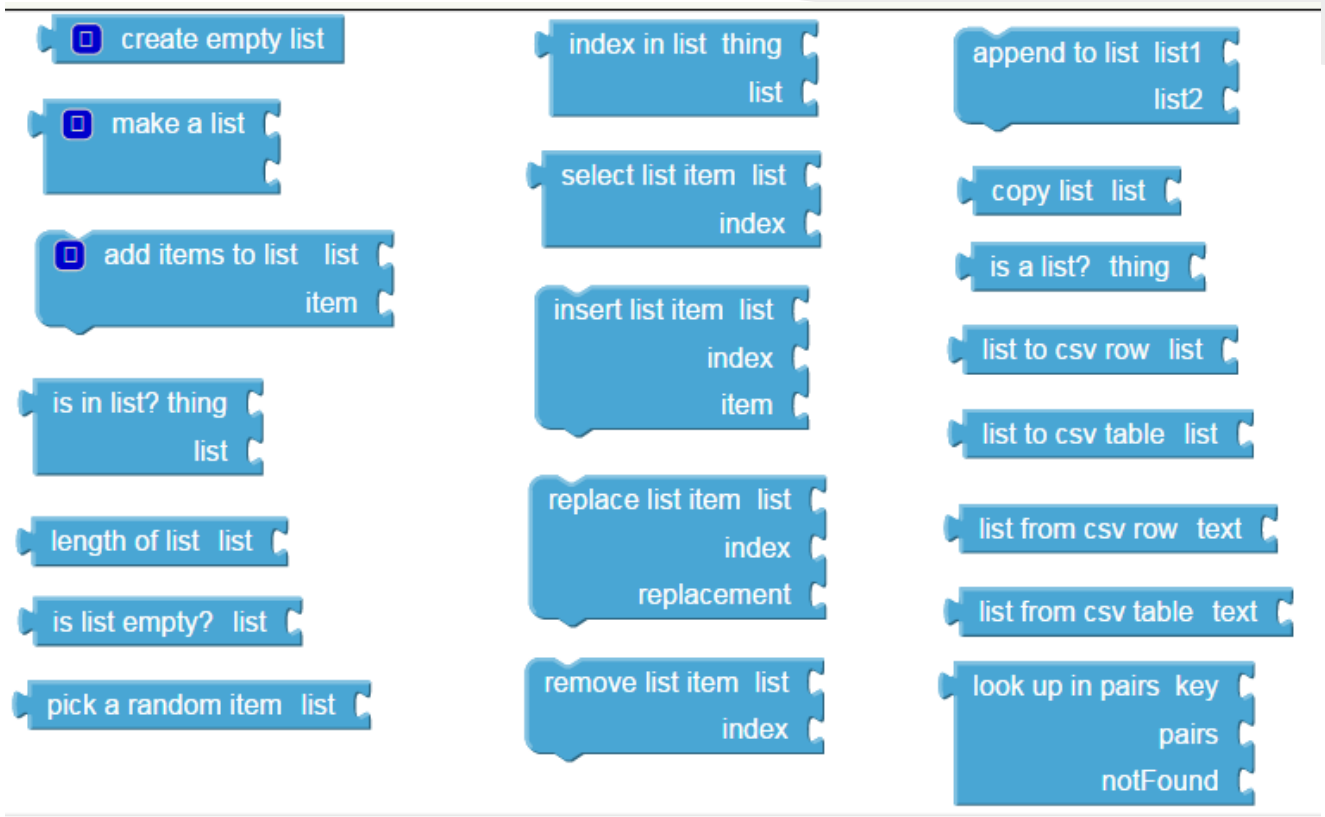

MIT App Inventor 2
Beta
Projects
Connect
Build
Help
Admin
My Projects
Guide
Report an Issue
Gallery
test@example.com
test
Screen1
Add Screen ...
Remove Screen
Designer
Blocks
Palette
User Interface
Layout
Media
Drawing and Animation
Ball
Canvas
ImageSprite
Sensors
Social
Storage
Connectivity
LEGO® MINDSTORMS®
Viewer
Display hidden components in Viewer
9:48
Screen1
Components
Screen1
Canvas1
Rename
Delete
Properties
Canvas1
BackgroundColor
White
BackgroundImage
None...
FontSize
14.0
LineWidth
2.0
PaintColor
Black
TextAlignment
center
Visible
showing
Width
Automatic...
Height
Automatic...

Blockly Editor



Problem with Current List Operators

Current List Operators



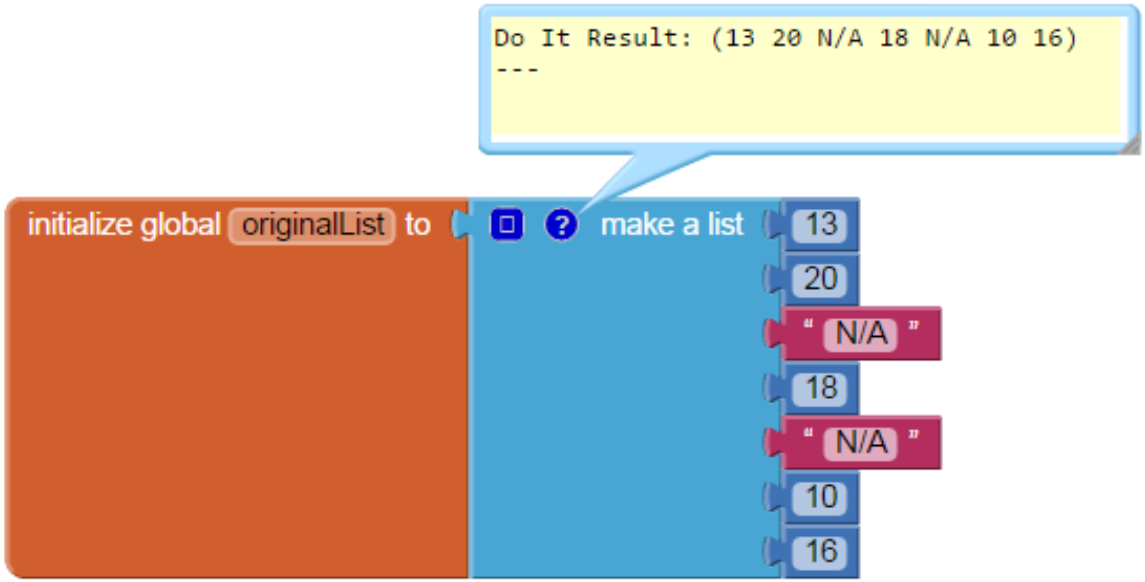
Map, Filter and Reduce

```
>>>(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
```

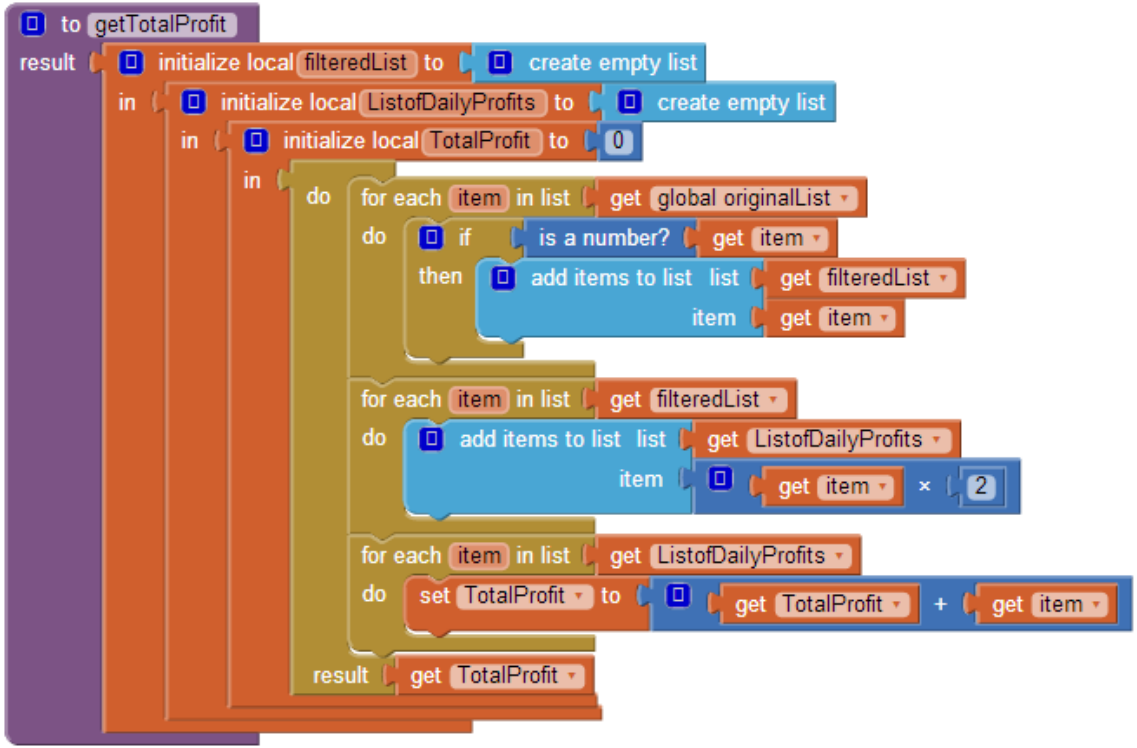
Berry's Lemonade

Date	# of lemonades sold	Daily Profit
6/1/13	13	
6/2/13	20	
6/3/13	N/A	
6/4/13	18	
.	.	
.	.	
.	.	
	Total Profit:	?

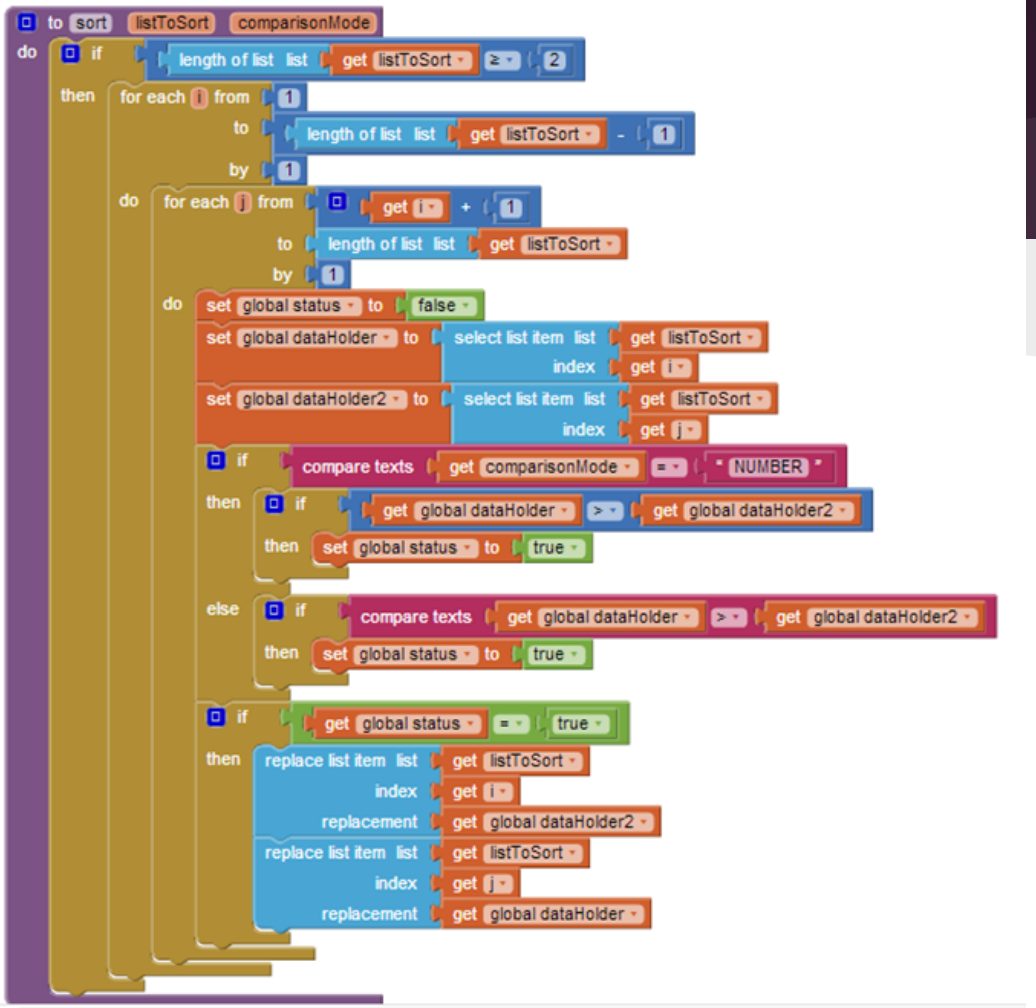
Old Design Using Loops



Old Design Using Loops



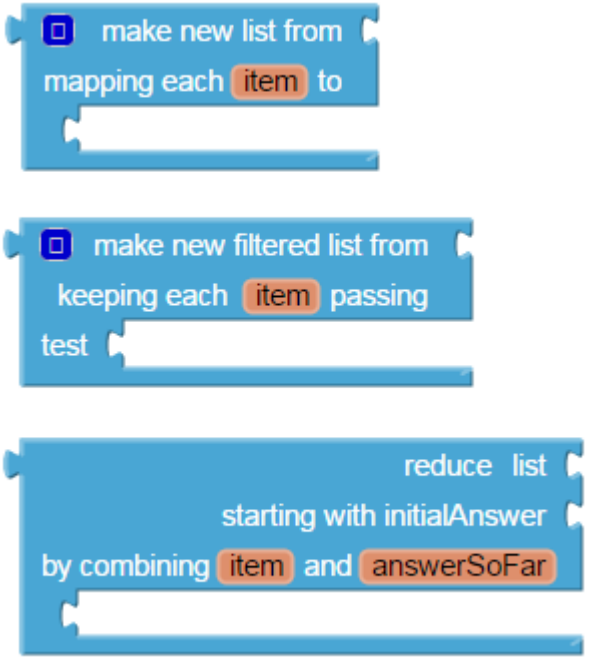
Sort: Old Design



http://www.imagnity.com/tutorials/app-inventor/list-sorting-on-app-inventor/

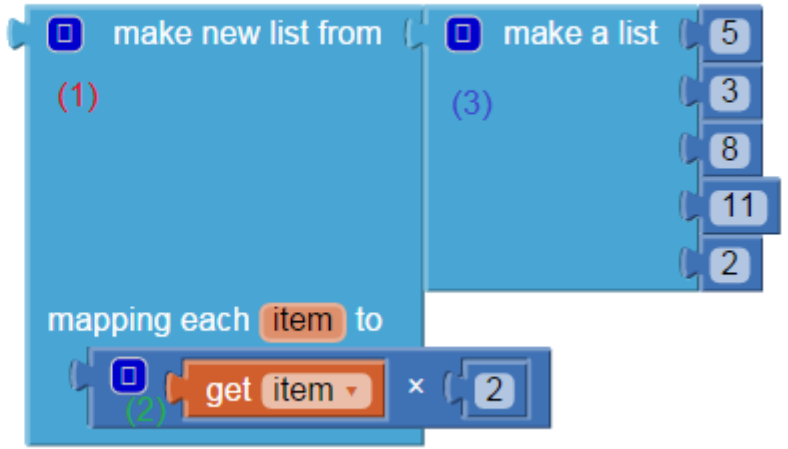
Solution: Addition of Higher-Order Operators

Map, Filter and Reduce

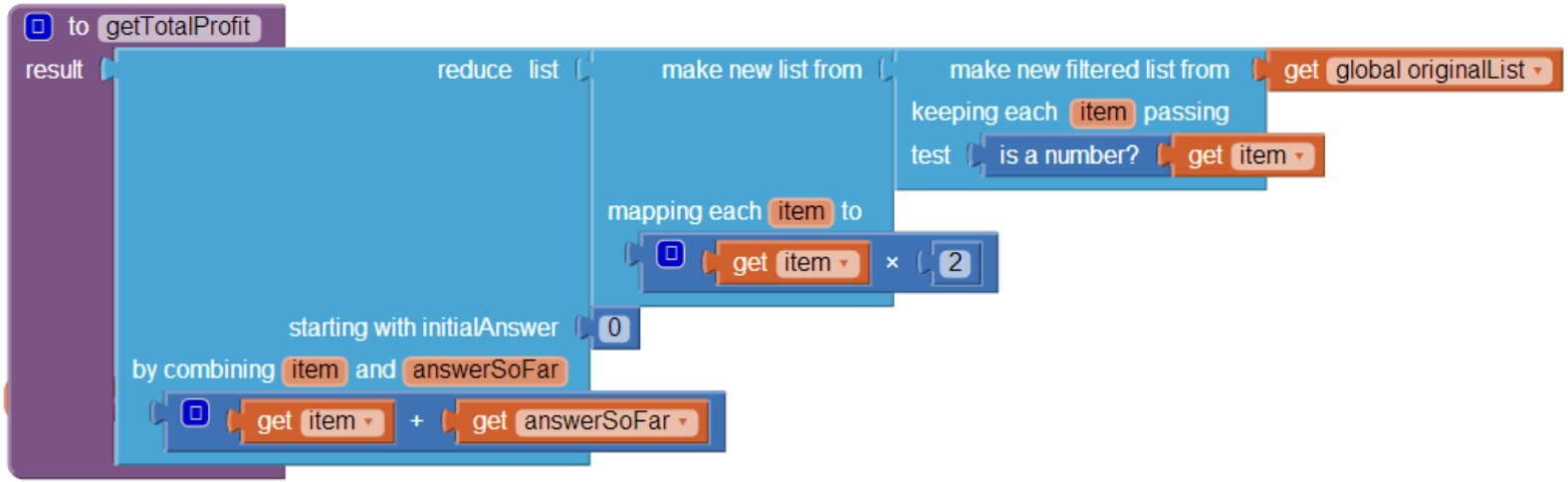


Map Block

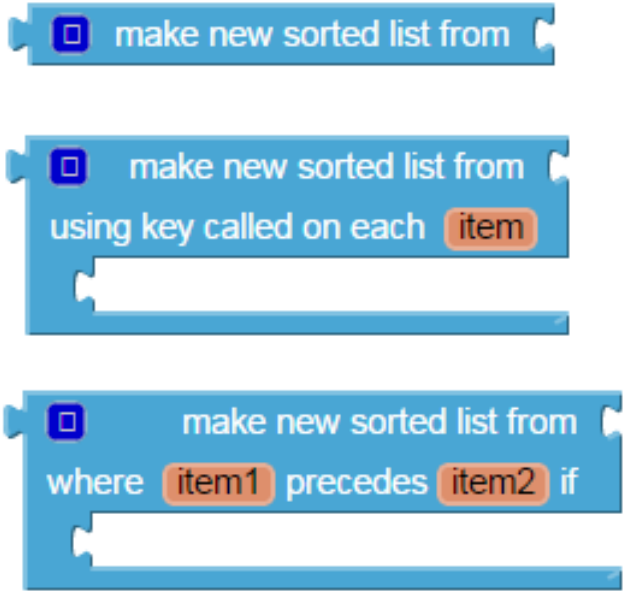
```
        (1)          (2)          (3)
map(lambda item: item * 2, [5,3,8,11,2])
```



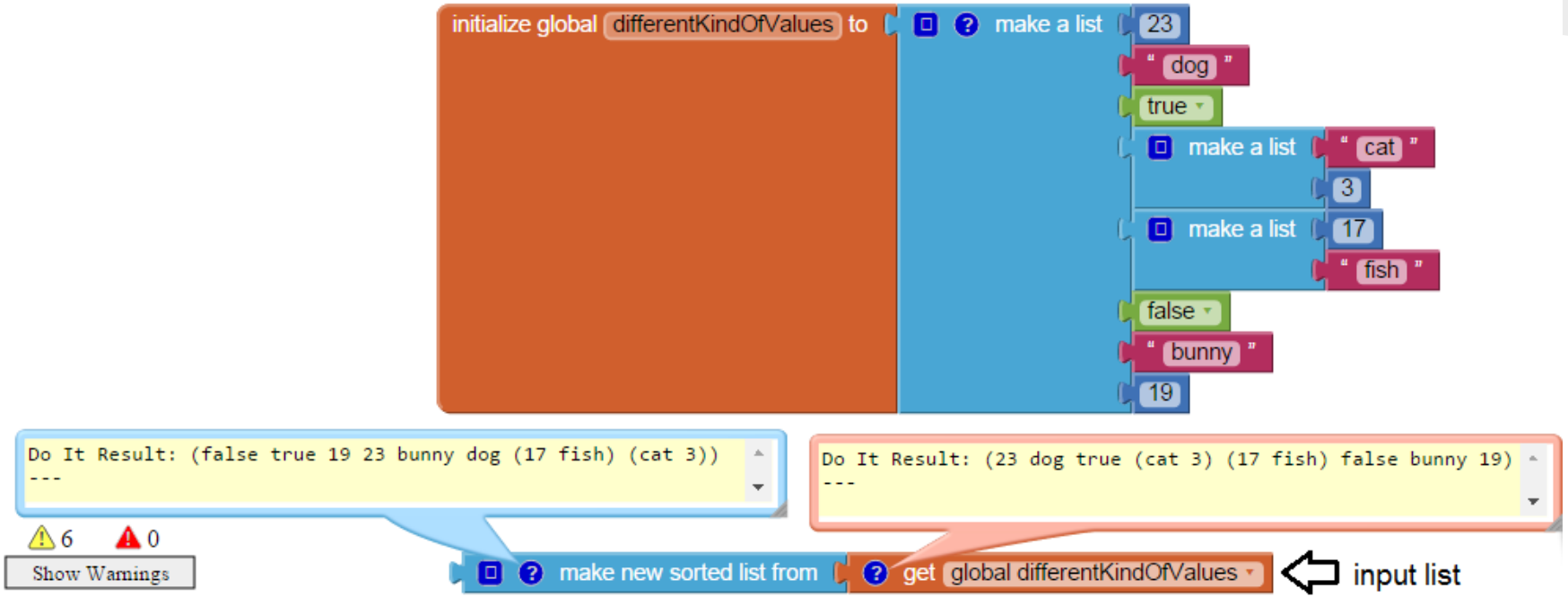
New Design Using Map, Filter and Reduce Operators



Three Sort Blocks



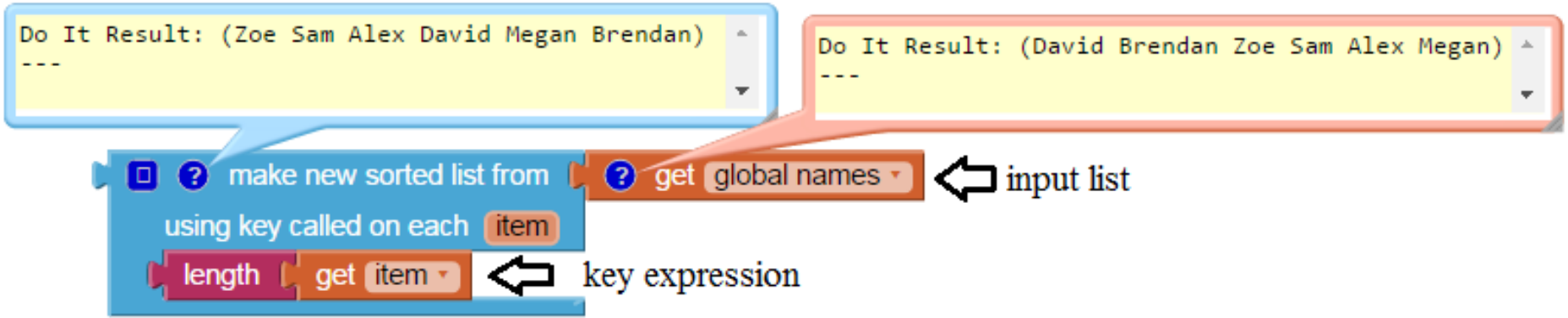
Basic Sort


initialize global differentKindOfValues to
make a list
23
" dog "
true
make a list
" cat "
3
make a list
17
" fish "
false
" bunny "
19
Do It Result: (false true 19 23 bunny dog (17 fish) (cat 3))

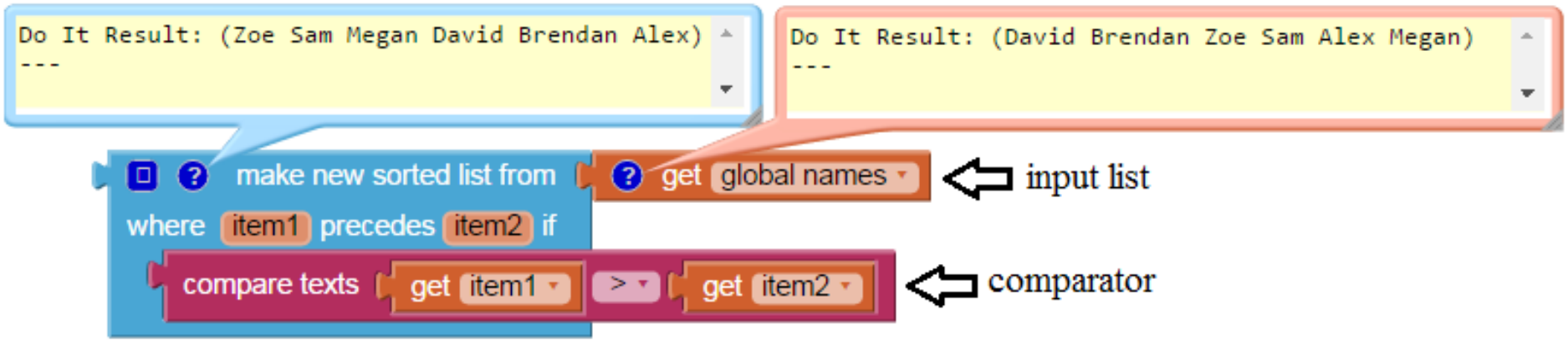
Do It Result: (23 dog true (cat 3) (17 fish) false bunny 19)

6
0
Show Warnings
make new sorted list from
get global differentKindOfValues
input list

Sort with Key

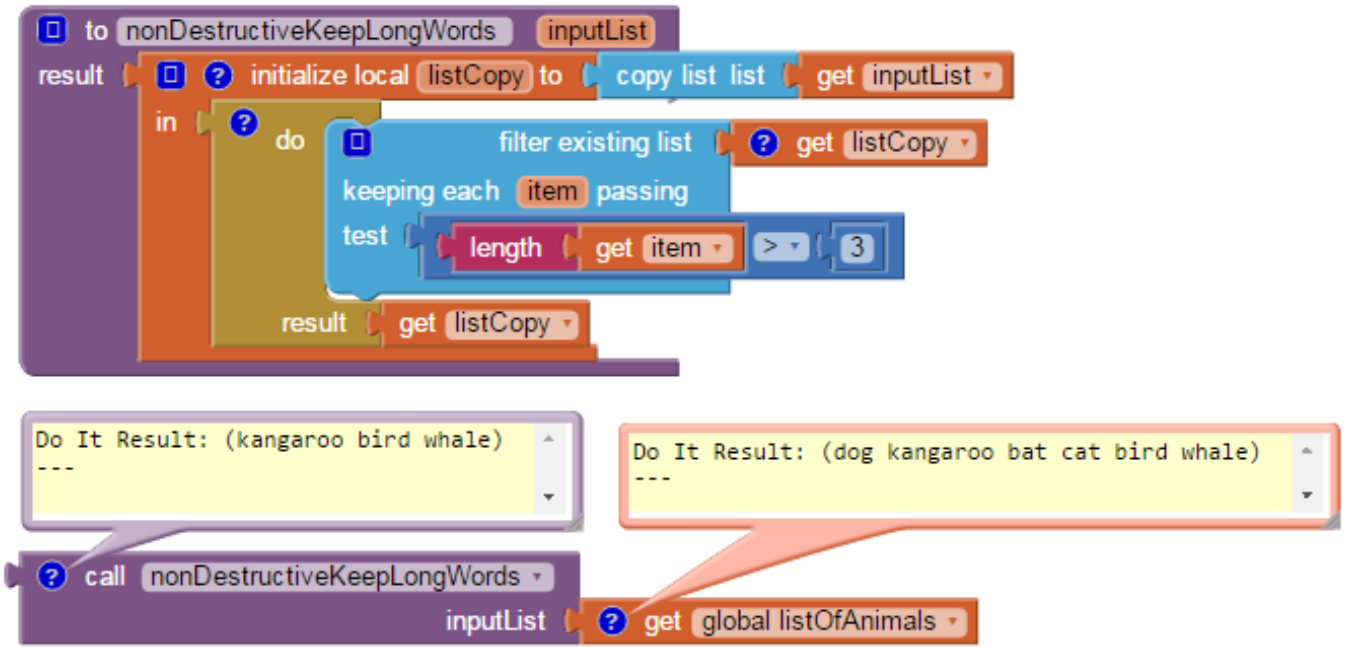


Sort with Comparator

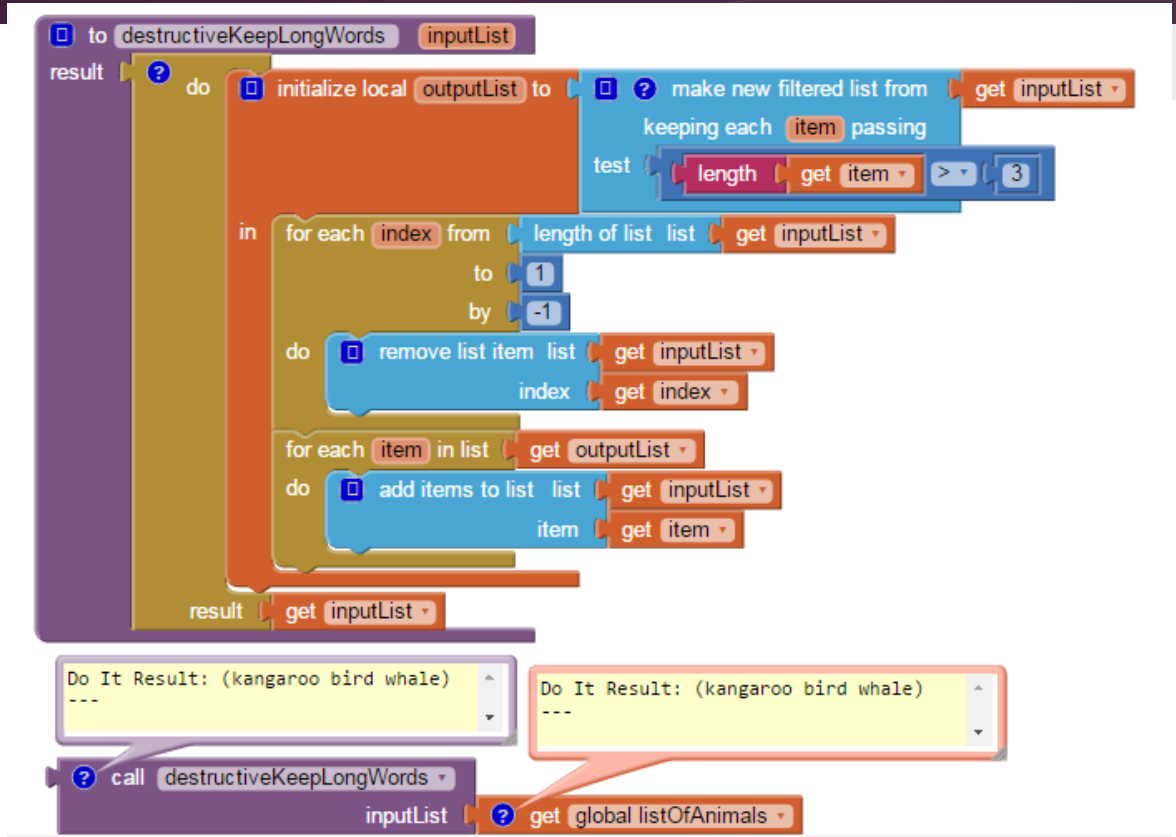


Destructive vs. Nondestructive Mechanism

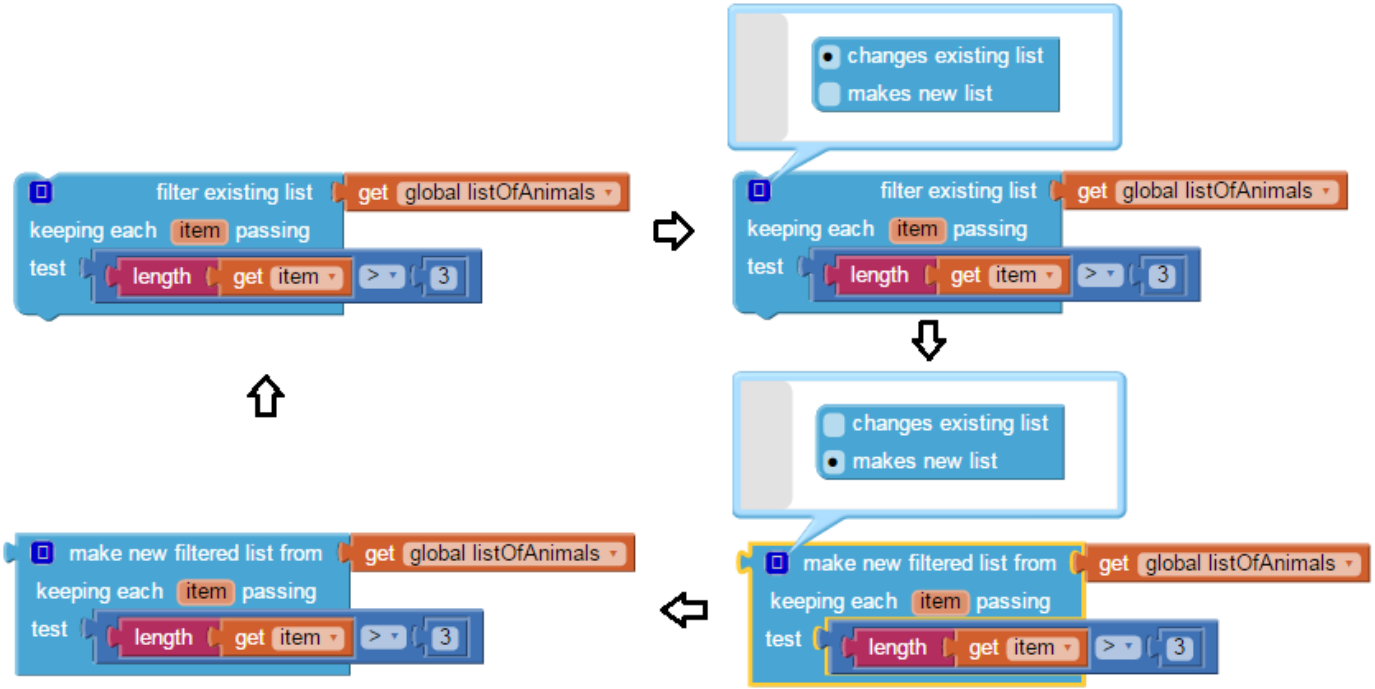
Simulating Nondestructive Version of a Destructive Filter Operator



Simulating Destructive Version of a Nondestructive Filter Operator



Destructive vs. Nondestructive Mechanism



Design and Results of User Study

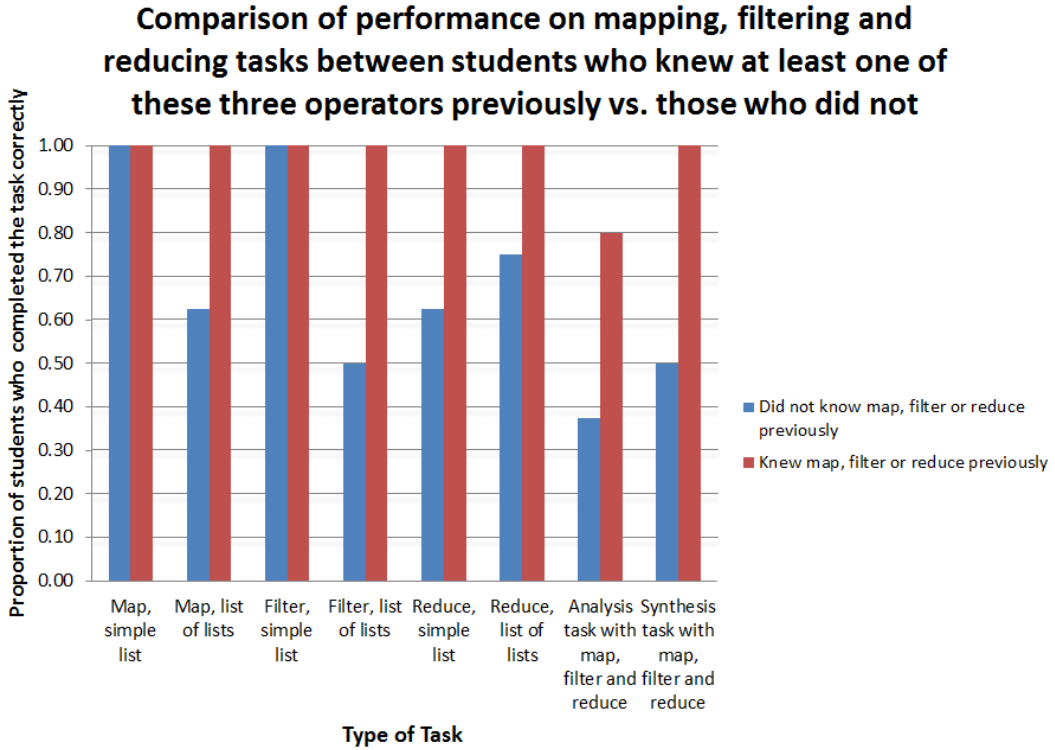
Design of User Study

- 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

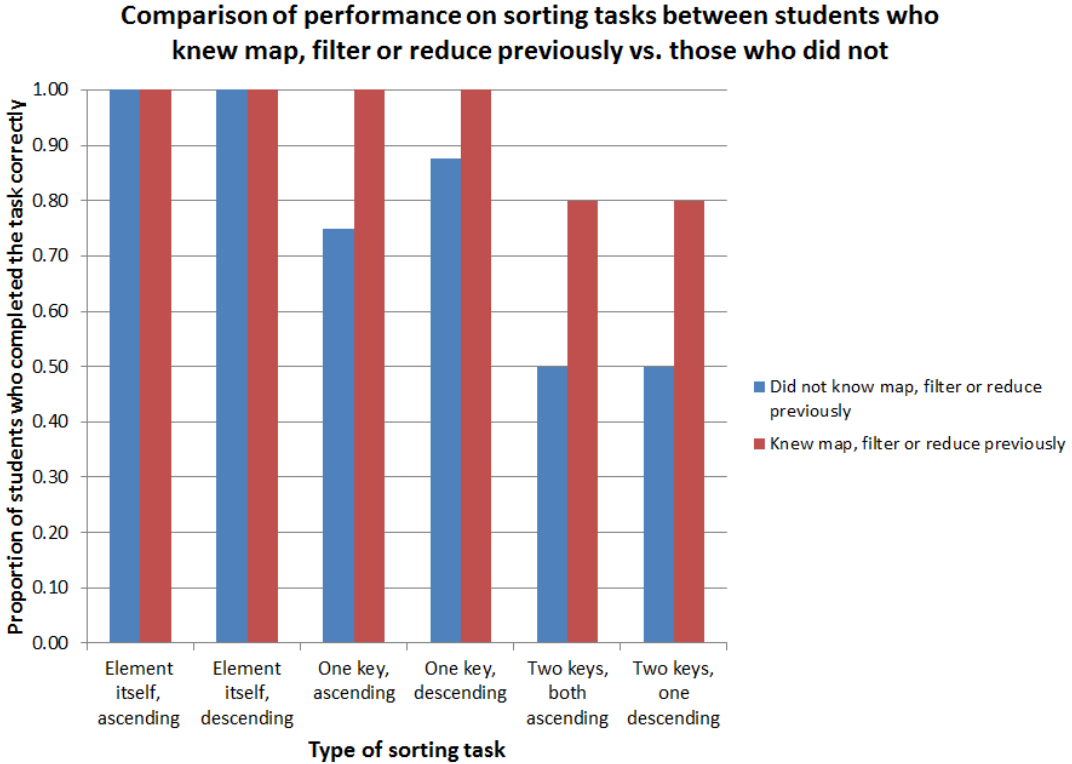
User Study Results Part 1



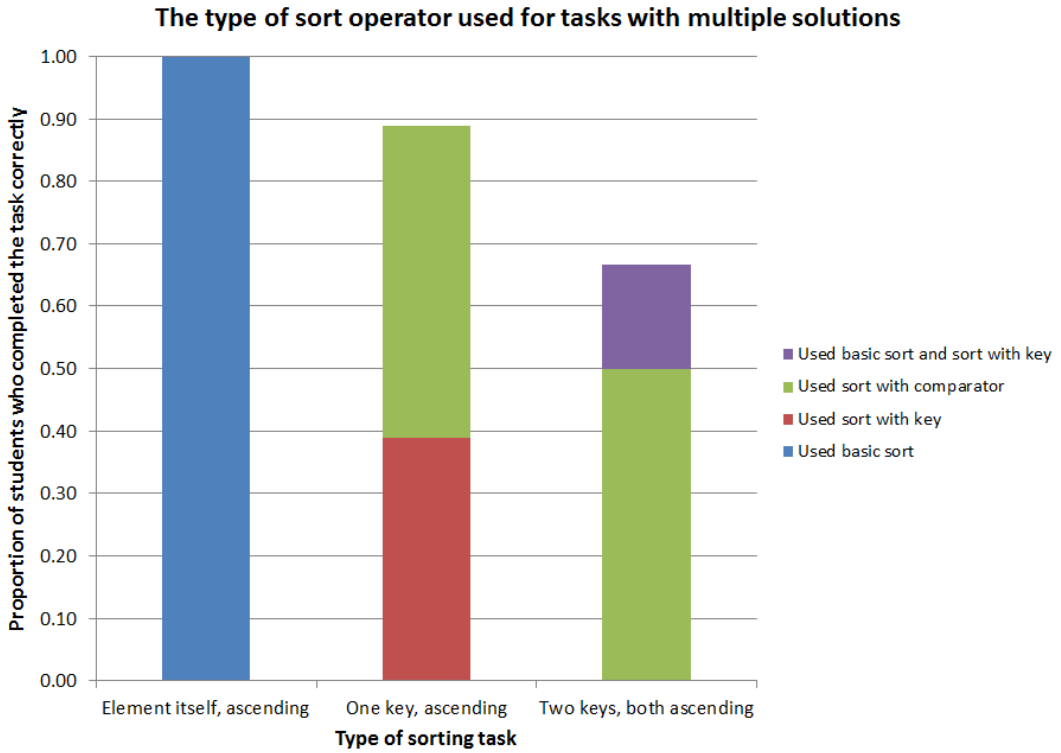
Feedback

- “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



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.”

Future Work

- finalizing the labels on the blocks
- additional user studies
  - compare loops vs. higher order operators
  - test destructive vs. nondestructive mechanism
  - study wider demographic base