

# 4 Programming



Peeking into Computer Science

© Jalal Kawash 2010

1

- Mandatory: Chapter 5 – Section 5.5
- Jalal's resources:
  - "How to" movies and example programs available at:  
<http://pages.cpsc.ucalgary.ca/~kawash/peeking/alice-how-to.html>
- JT's resources:
  - [www.cpsc.ucalgary.ca/~tamj/203/extras/alice](http://www.cpsc.ucalgary.ca/~tamj/203/extras/alice)



## Reading Assignment

Peeking into Computer Science

© Jalal Kawash 2010

2

- A list is a variable that can be treated as one entity.

Student 1	Student 2	Student 3	Student 4	Student 5
-----------	-----------	-----------	-----------	-----------

Example: I need to print out the whole class list

- ...but also a list consists of elements and the individual elements can be accessed separately.

Student 1	Student 2	Student 3	Student 4	Student 5
-----------	-----------	-----------	-----------	-----------

Example: I need to fix this one student's grade



## JT's Extra: What Is A List?

Peeking Into Computer Science

© Jalal Kawash 2010

3

At the end of this section, you will be able to:

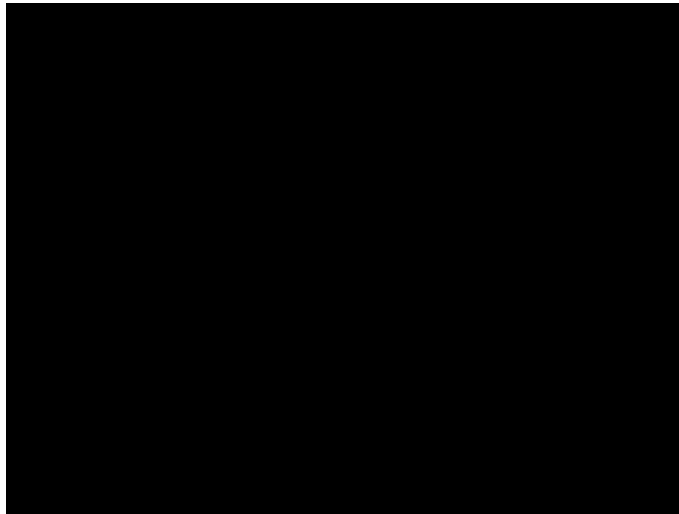
1. Understand the list structure
2. Create lists in Alice
3. Apply the *for all in order* construct to lists
4. Apply the *for all together* construct to lists
5. Use functions to collect user input



## Objectives

Peeking into Computer Science

© Jalal Kawash 2010



## A list of Astronauts in Alice

- A list is a tuple of numbered (indexed) objects
- A list is a variable



## Lists

## Creating a List in Alice

Peeking into Computer Science © Jalal Kawash 2010 7

**March 3 soldiers forward**

**Without lists: you would create three soldier objects**

**Example 12: 'How not to' approach**

The actions for each list would have to be repeated three times (once per object)

A list would allow the soldiers to be grouped together and instructions wouldn't have to be repeated explicitly

## JT's Extra: What If A List Weren't Used?

Peeking into Computer Science © Jalal Kawash 2010 8

The image shows a Scratch code editor with two examples of loops. The first example is a 'For all in order' loop with three actions: 'play sound cow.moo (101.56 /)', 'say Forward march!', and 'move forward 3 meters'. The second example is a 'For all together' loop with three actions: 'play sound world.right face (0:02.500)', 'say Right...face!', and 'turn right 0.25 revolutions'. Red arrows point to the 'For all in order' and 'For all together' options in the top menu. Red brackets group the actions in each loop, with red text explaining the difference: 'FOR ALL IN ORDER: Each list element carries out instructions one element-at-a-time' and 'FOR ALL TOGETHER: The instructions are performed at the same time by each list element'.

**FOR ALL IN ORDER**  
Each list element carries out instructions one element-at-a-time

**FOR ALL TOGETHER**  
The instructions are performed at the same time by each list element

**JT's Extra: For all in order and together**

Peeking into Computer Science © Jalal Kawash 2010 9

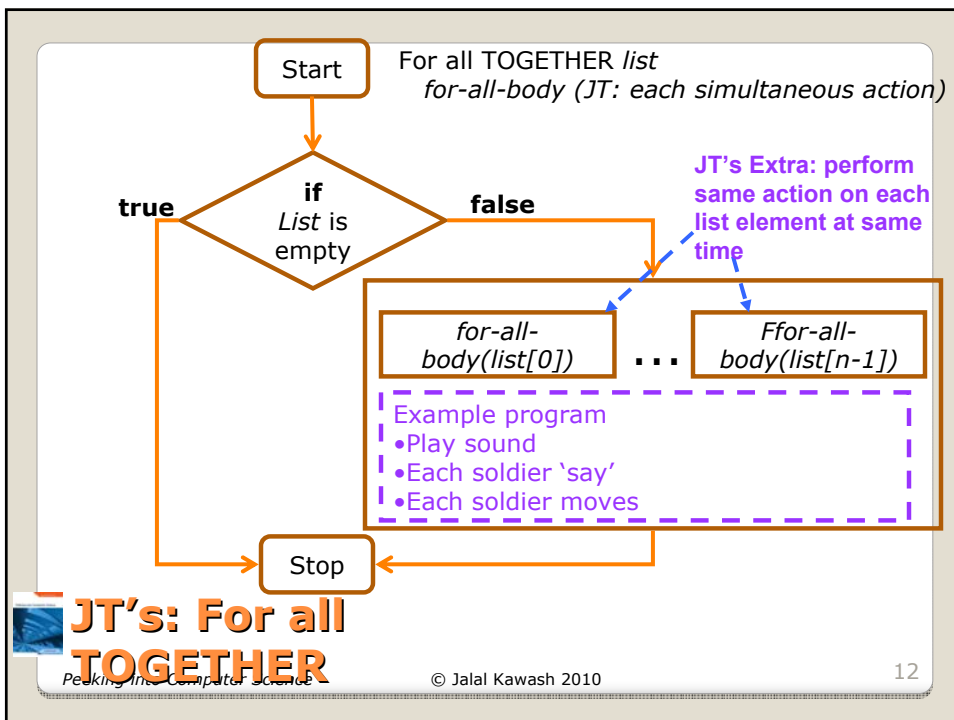
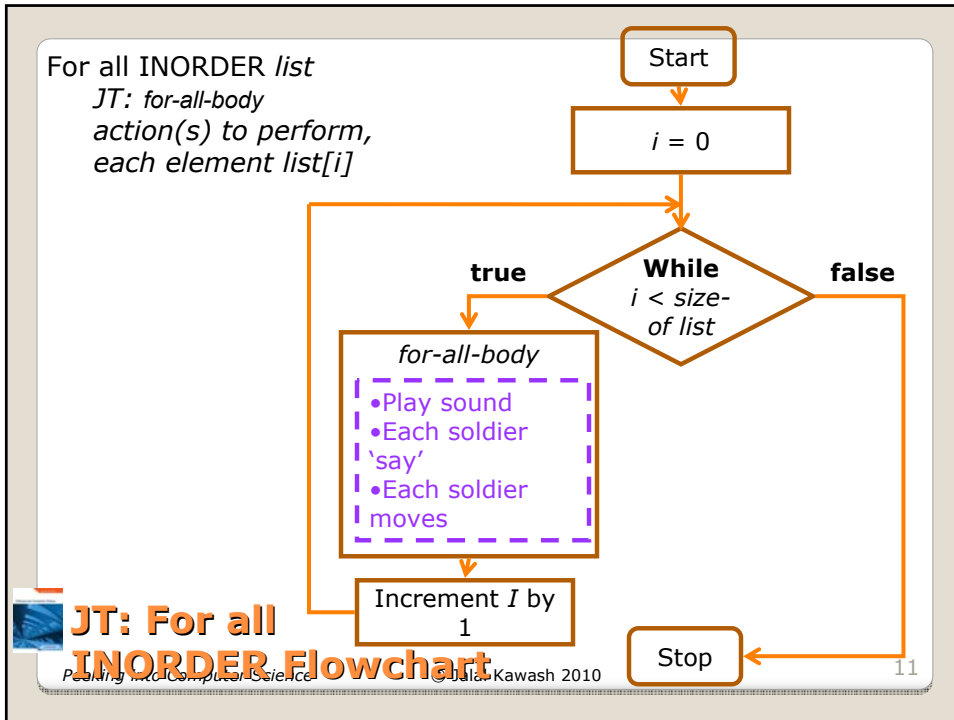
**Example 13:**  
Lists, for all in order, for all together

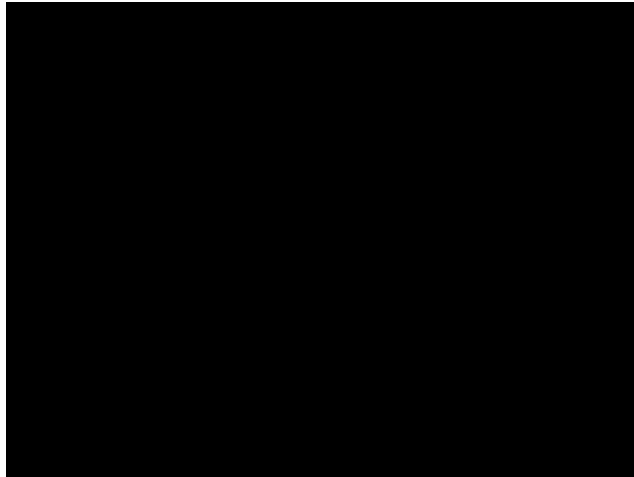
For all in order: body of loop repeats three actions for each list element (soldier) one at-a-time

For all together: body of loop repeats three actions for each list element (soldier) simultaneously

**JT's Extra: March Example (Soldiers Again)**

Peeking into Computer Science © Jalal Kawash 2010 10





## More Complex Example

```
Loop rounds times ▾ time show complicated version
  ▾ For all world.spaceTeam ▾ , one Obj item_from_spaceTeam at a time
    item_from_spaceTeam ▾ turn backward ▾ 0.25 revolutions ▾ duration = 0.25 seconds ▾ more... ▾
    item_from_spaceTeam ▾ turn forward ▾ 0.25 revolutions ▾ duration = 0.25 seconds ▾ more... ▾
  ▾ For all world.spaceTeam ▾ , every Obj item_from_spaceTeam together
    item_from_spaceTeam ▾ move up ▾ 1 meter ▾ duration = 0.5 seconds ▾ more... ▾
    item_from_spaceTeam ▾ roll left ▾ 1 revolution ▾ duration = 0.5 seconds ▾ more... ▾
    item_from_spaceTeam ▾ move down ▾ 1 meter ▾ duration = 0.5 seconds ▾ more... ▾
  ▾ For all world.spaceTeam ▾ , one Obj item_from_spaceTeam at a time
  ▾ For all world.spaceTeam ▾ , every Obj item_from_spaceTeam together
```

## world.dance Method

Peeking into Computer Science

© Jalal Kawash 2010

14

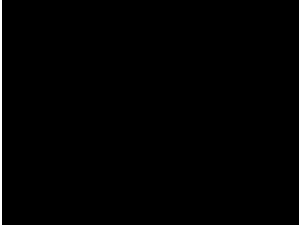
Loop rounds times ▾ time show complicated version

For all world.spaceTeam ▾, one Obj item\_from\_spaceTeam at a time

- item\_from\_spaceTeam ▾ turn backward ▾ 0.25 revolutions ▾ duration = 0.25 seconds ▾ more... ▾
- item\_from\_spaceTeam ▾ turn forward ▾ 0.25 revolutions ▾ duration = 0.25 seconds ▾ more... ▾

For all world.spaceTeam ▾, every Obj item\_from\_spaceTeam together

- item\_from\_spaceTeam ▾ move up ▾ 1 meter ▾ duration = 0.5 seconds ▾ more... ▾
- item\_from\_spaceTeam ▾ roll left ▾ 1 revolution ▾ duration = 0.5 seconds ▾ more... ▾
- item\_from\_spaceTeam ▾ move down ▾ 1 meter ▾ duration = 0.5 seconds ▾ more... ▾



## world.dance Method

Peeking into Computer Science © Jalal Kawash 2010 15

For all INORDER list  
for-all-body

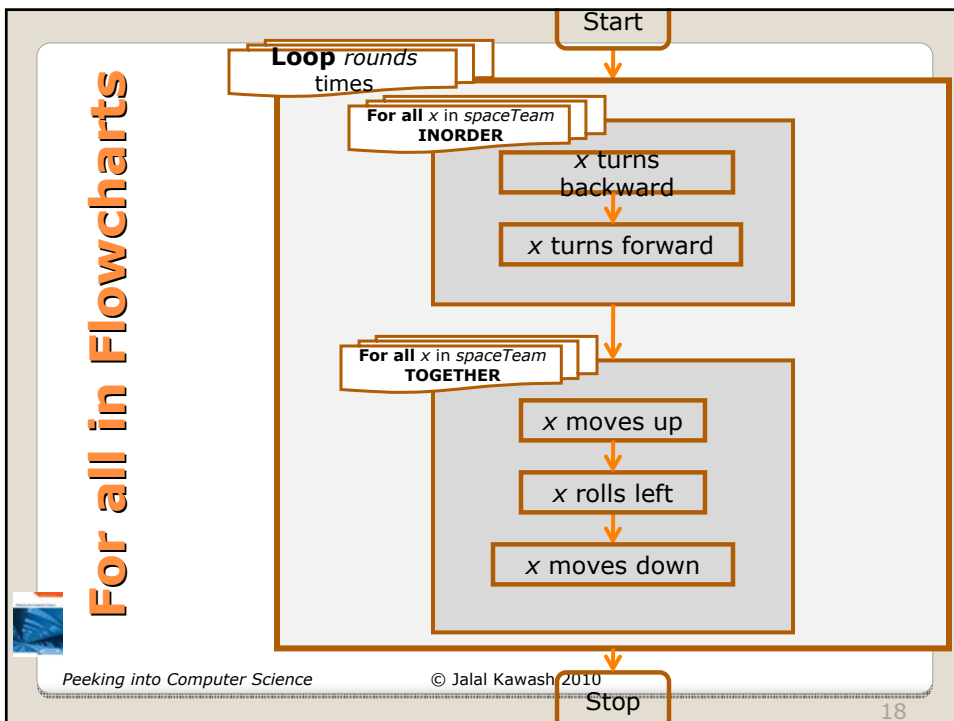
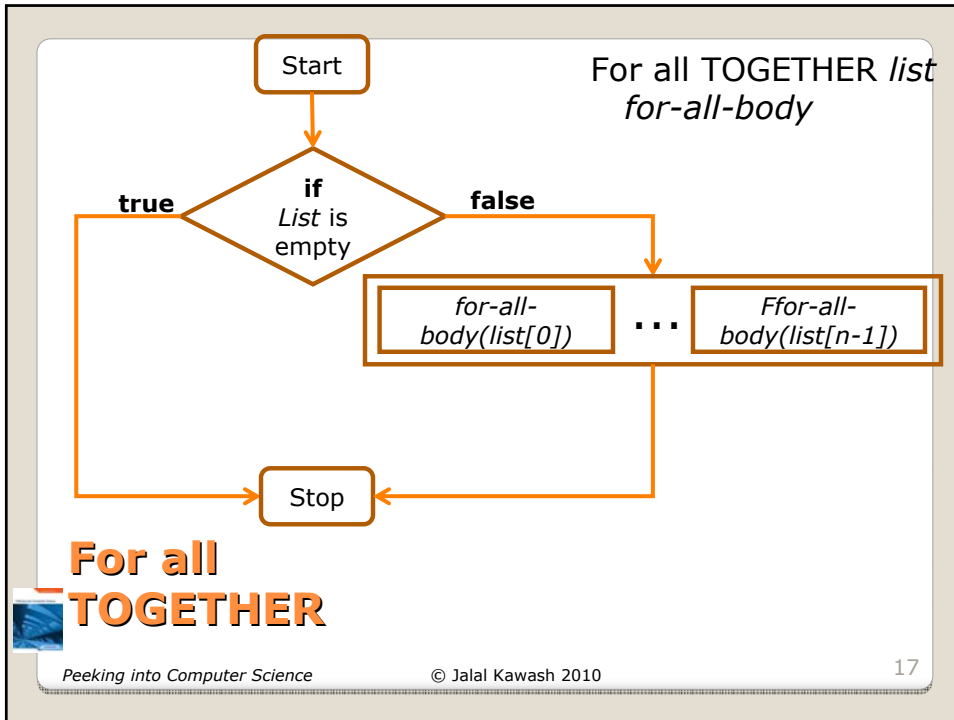
```

graph TD
    Start([Start]) --> I0[i = 0]
    I0 --> While{While  
i < size-  
of list}
    While -- true --> FB[for-all-  
body(list[i])]
    FB --> Inc[Increment I by  
1]
    Inc --> While
    While -- false --> Stop([Stop])
  
```

## For all INORDER

Peeking into Computer Science © Jalal Kawash 2010 16





The screenshot displays a programming environment. On the left, a 'world's details' window shows a tree view with 'properties', 'methods', and 'functions' tabs. Under 'methods', there are several items: 'what as a string', 'ask user', 'ask user for a number', 'ask user for yes or no', and 'ask user for a string'. Below this, a 'world.dance' object is selected, showing 'world.my first method' with 'No parameters' and 'No variables'. The code editor shows a snippet: `world.dance rounds = ask user for a number question = Enter the number of rounds: more...`. On the right, a 'Question' dialog box is open, asking 'Enter the number of rounds:' with a text input field containing the number '3' and 'OK' and 'Cancel' buttons.

## Collecting User Input