

# Technical English II

Computing with *J*

Steven Phillips, PhD

Neuroscience Research Institute  
National Institute of Advanced Industrial Science and Technology (AIST)  
([steve@ni.aist.go.jp](mailto:steve@ni.aist.go.jp))

# Write down this web page:

<http://staff.aist.go.jp/steven.phillips>

- Click on [Technical English II](#) for slides
- Click on [Participants for Experiments](#) (アルバイト)

# Course Objectives

**Experience** at listening to technical English

**Experience** at giving a presentation in *English*

Ability to do simple programming (in *J*)

**... practice, practice, practice, ...**

# Course Outline (Chiteki + Kino)

Week 1: Lecture (1) – Introduction

Week 2: Lecture (2) – J language

Week 3: Lecture (3) – J language

Week 4: **Student Presentation**  
**(groups 1-5)**

Week 5: **Student Presentation**  
**(groups 6-10)**

# Course Outline (Kankyo + Energy)

Week 1: Lecture (1) – Introduction

Week 2: Lecture (2) – J language

Week 3: Lecture (3) – J language

Week 4: Prepare presentation

Week 5: **Student Presentation 1:**  
**(short version): groups 1-10**

Week 6: Lecture (4) – J language

Week 7: Lecture (5) – J language

Week 8: Lecture (6) – J language

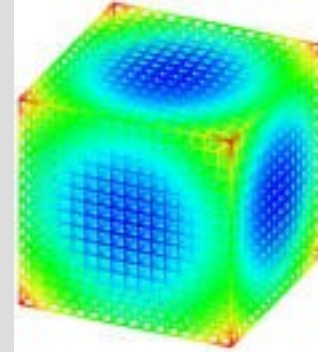
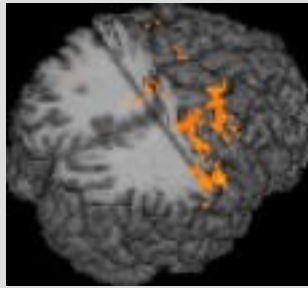
Week 9: **Student Presentation 2**  
**(long version): groups 1-5**

Week 10: **Student Presentation 2**  
**(long version): groups 6-10)**

# Lecture 1: Outline

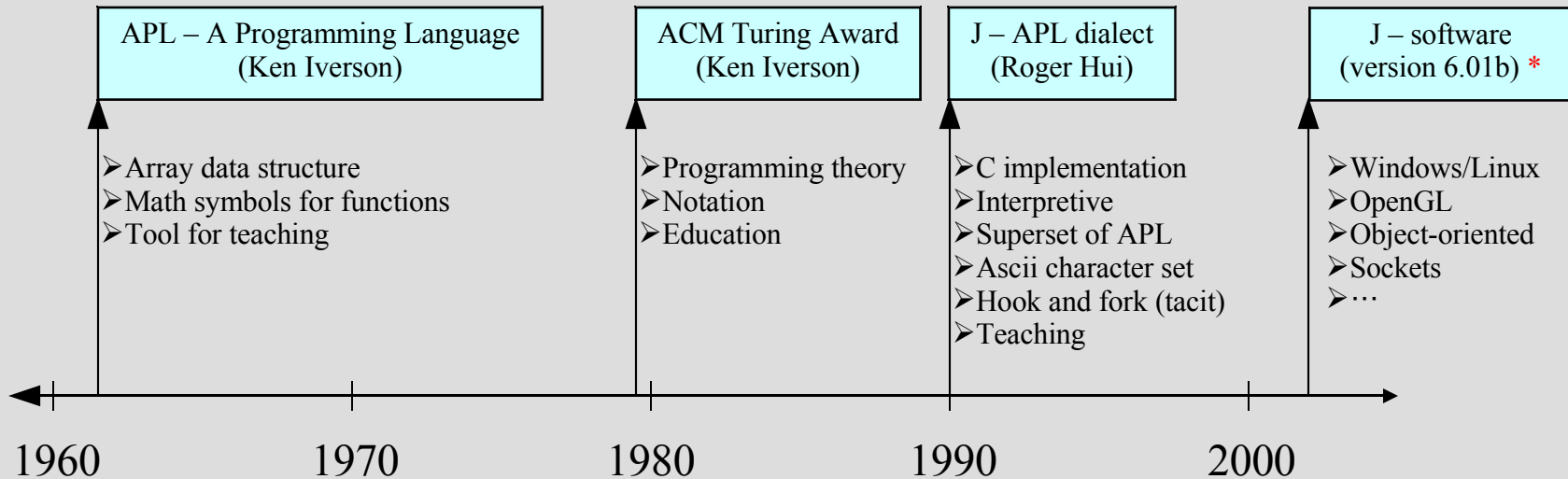
1. Introduction    demonstration of J system
2. Main topic    some details of J
3. Closing    course and exam (student presentation)

## 2. The “J” language



*J* is an *array-oriented* programming language

# 2.1 History of J



Applications (large array-structured data):

- ✓ mathematical/scientific computation
- ✓ simulation
- ✓ business

**\* J Software home page**

# Demo of J system

- Calculator
- General (functional/array) programming
- Image manipulation
- OpenGL
- Graphical User Interface
- Games, simulations, ...

## 2.2 Using *J* ?

*J* has some *unusual* features:

- easy to get started
- powerful, executable math-like expressions
- **Free!**

## 2.2.1 Getting *J* (windows)

If *not* installed on your machine, goto:

- <http://www.jsoftware.com/>
- install software

## 2.2.2 Starting *J* (windows)

- Find the file *j.exe* (usually on desktop, C: drive)
- *Double click* file to start J system
- *Press F1* to get help on J functions

# 2.2.3 J environment

Interface windows

Execution

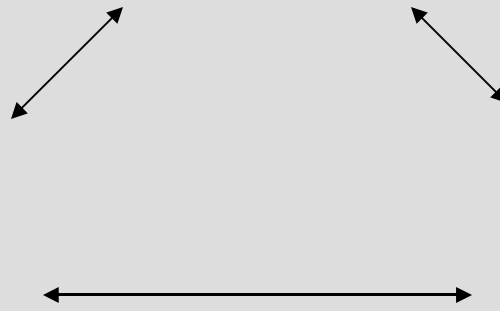
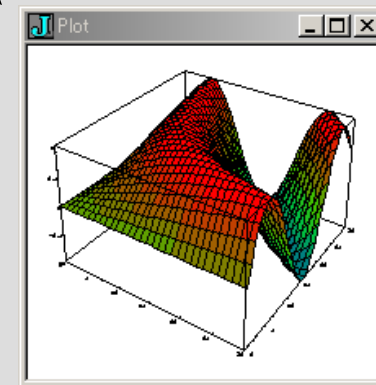
Version: 6.01beta  
OS: **Windows, Linux, Mac**  
License: **Free** (full version)

Script

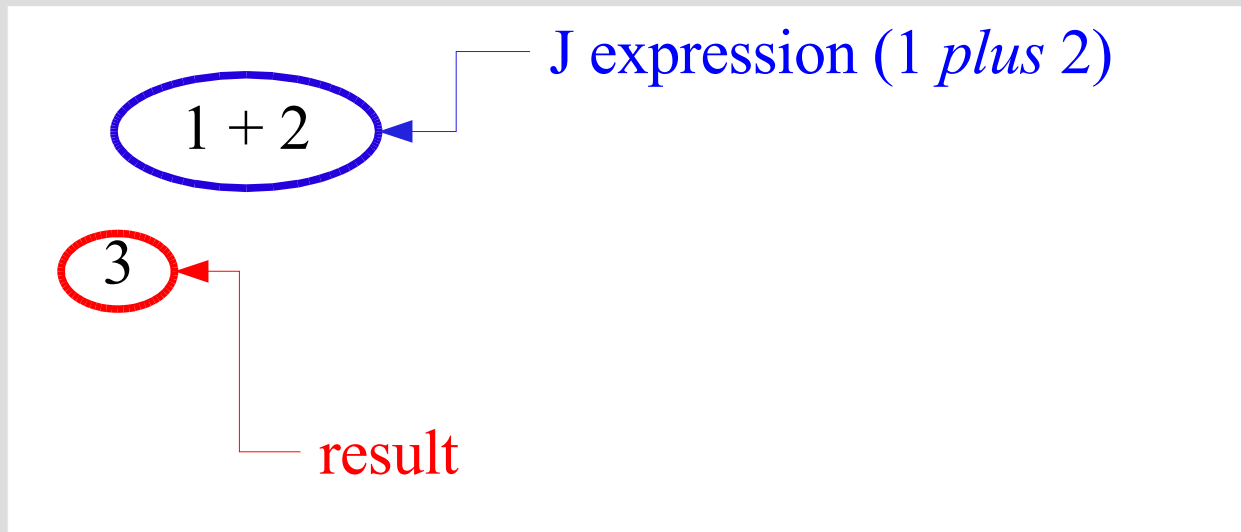
Generated  
(e.g., Plot)

```
J 1.ijx
File Edit Run Tools Studio Window Help
NB. J execution window
load 'plot numeric trig'
sin steps 0 3 30
0 0.0998334 0.198669 0.29552 0.389418 0
'surface' plot sin */~ steps 0 3 30
```

```
J trig.ijs
File Edit Run Tools Studio Window Help
NB. J script window
NB. trigonometric functions
sin=: 1&o.
cos=: 2&o.
tan=: 3&o.
sinh=: 5&o.
```



## 2.2.4 Example



## 2.2.5 More examples

$2 * 5$   
10

multiplying two numbers

$10 * 1\ 2\ 3$   
10 20 30

scaling a vector

$10\ 20\ 30 * 1\ 2\ 3$   
10 40 90

multiplying two vectors (item-by-item)

$10\ 20 */ 1\ 2\ 3$   
10 20 30  
20 40 60

vector cross-product gives matrix

## 2.2.6 *J* arrays

1

*scalar*, 0 **axes**

→  
1 2 3 4

*vector*, 1 **axis** of length 4

→  
1 2 3 4  
↓  
5 6 7 8

*matrix*, 2 **axes** of length 2 and 4

↙  
→  
↓

*r-array*, *r* **axes** (*r* is the *rank*)

## 2.2.7 Some *J* functions

$+$  ,  $-$  ,  $*$  ,  $\%$   
 $+\ :$  ,  $-\ :$  ,  $*\ :$  ,  $\%\ :$

$\{.$  ,  $\{\ :$   
 $|.$  ,  $|\ :$

and more ...

Arithmetic:

*add, subtract, multiply, divide*

*double, halve, square, square root*

Structural:

*first, last item in array*

*reverse order array items, axes*

## 2.3 Why *J*?

A small set of *primitive* symbols combine to produce many functions

## 2.3.1 Example: *average*

Subjects	Michiko	Mizuki	Yuu	Ryo	Nozomi	Average
<b>Scores</b>	7	9	8	6	6	7.2

Definition (*average*)

English

**The *sum of all the scores divided by the number of scores***

*J* words

**(add all divide count) Scores**

*J* symbols

**(+ / % #) Scores**

## 2.3.4 *J* versus *C*

*J* code (functional)

```
(+ / % #) Scores
```

```
(+ / % #) "1 Scores
```

*C* code (procedural)

```
for (i=0; i<NumStud; ++i){  
    Sum += Scores[i]  
}  
Mean = Sum / NumStud
```

```
for (j=0; j<NumSubj; ++j){  
    for (i=0; i<NumStud; ++i){  
        Sum[j] += Scores[i][j]  
    }  
    Mean[j] = Sum[j] / NumStud  
}
```

# 3. Student Presentation

## English presentation practice (of **J** “functions”)

- Groups of 2-4 students (10 groups)
- Present one J “function” (e.g., **+**: double)
- Format:
  - Presentation 1 (week 5): 5 minutes + questions
  - Presentation 2 (week 9/10): 10 minutes + questions

## Web links:

- **J** software (<http://www.jsoftware.com>)
- **J** in Japan [HTML](#)

# 3.1 Example topic: Group 1

- Group 1 will give a short 10 minute presentation that explains the *function (meaning)* of the J symbol “|” (*magnitude/residue*)
- When applied *monadically* (i.e.,  $| y$  – to number  $y$  on the right hand side), this function returns the *size* of  $y$ :
  - Try  $| 4$  (applied to a positive integer)
  - Try  $| _3$  (applied to a negative integer)
  - Try  $| 3j4$  (applied to a complex number)
- When applied *dyadically* (i.e.,  $x | y$  – to number  $x$  on the left side and  $y$  on the right side), this function returns the remainder of  $y$  after removing an “whole” number of  $x$ 's:
  - Try  $3 | 7$  (returns the remainder of 7 after removing an integral number of 3's)
  - Try  $3 | _7$  (explain other cases, such as real numbers, 0, and 1)
- Group 1 will give their presentation during **Lecture 4**

## 3.2 Group schedule

Group	J symbol	Pronunciation	Monadic	Dyadic	Lecture
1		<i>Bar</i>	Magnitude	Residue	4
2	{.	<i>Left-brace-dot</i>	Head	Take	4
3	}.	<i>Right-brace-dot</i>	Behead	Drop	4
4	{	<i>Left-brace-dot</i>	Catalogue	From	4
5	i.	<i>I-dot</i>	Integers	Index-of	4
6	<.	<i>Less-than-dot</i>	Floor	Lesser-of	5
7	.	<i>Bar-dot</i>	Reverse	Rotate	5
8	:	<i>Bar-colon</i>	Transpose	Transpose	5
9	#	<i>Hash</i>	Tally	Copy	5
10	,	<i>Comma</i>	Ravel	Append	5

## 3.2 Additional information (web page)

- Course schedule
- Student presentation – format and assessment
- Student presentation topics – J functions
- Glossary of J terms – please read carefully
- **Your group assignment – details**

# Summary

## Course:

- Objective – Practice at English presentation
- Topic – *J* (computing language)
- Exam – student presentation
- Materials – <http://staff.aist.go.jp/steven.phillips>