SI413: Programming Languages and Implementation

Ethan Panal and Taylor Cooper

This course examines basic concepts underlying the design of modern programming languages: types, control structures, abstraction mechanisms, inheritance, and constructs for programming. This course will include programming assignments in several languages.

J Language

- J is a mathematical language based on the APL language and invented by Kenneth Iverson and Roger Hui
- J language is **terse** but *powerful*
- J is used by several corporations such as Hewlett Packard and Intel

J Term	Other Language Term/Concept
Verb	Function or operator
Noun	Object or variable or constant
Copula	Assignment
Punctuation	Separator
Adverb	n/a
Conjuction	n/a
Sentence	Executable unit

Table from "A Casual J Tutorial" http://jeffzellner.com/miidaj/

Defining Features

- Array based programming of J allows for loopless code.
- **Verbs** are short rules that are applied to an array from right to left
- **Nouns** are objects such as integers, that verbs operate on.
- There are two kinds of verbs, **monads** and **dyads**. Dyads have arguments before and after the verb while monads are only followe by a noun.
- Monads and Dyads change the meaning of verbs which allow for more ways objects/nouns in arrays can be manipulated.

Example J language:

```
run=: 2 2  $ 1 2 3 4    <-- '$' creates a 2x2 matrix
                            named 'run' and fills it
                            with the integers 1,2,3,4
```

```
   ^run              <--- monad form of the verb '^'
2.71828 7.38906      <--- for every object in 'run' y,
20.0855 54.5982           e^y is outputed.
```

```
   run^2       <-- dyad form of the verb '^'
1  4           <-- use of this verb takes every
9 16               object of 'run' to the second
                   power to output 1,4,9,16
```