



University of Auckland

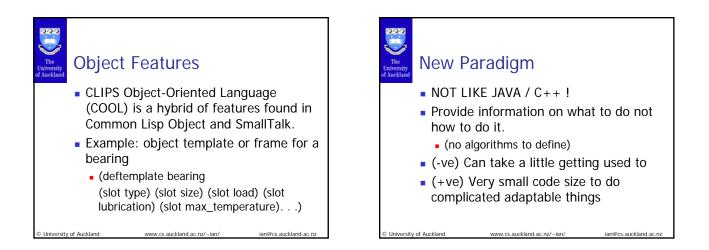
Overview

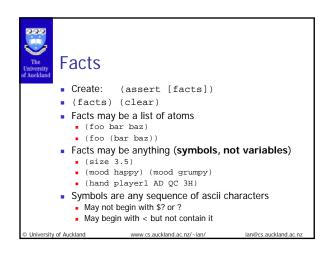
- CLIPS is a programming language that provides support for rule-based, objectoriented and procedural programming.
- The search in the Inference Engine uses forward chaining and rule prioritization.
- Looks like LISP (List Processing) with object features

www.cs.auckland.ac.nz/~ian/

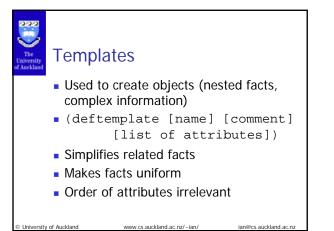
ian@cs.auckland.a

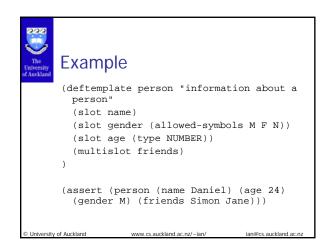
Developed by NASA in the 80s

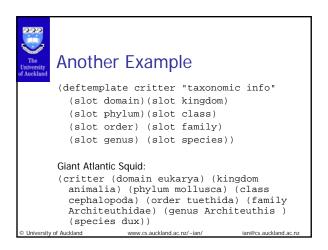


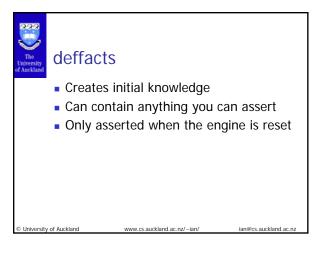


The University f Auekland	Example	e	
	(assert	(person name I	Daniel))
	(assert	(age Daniel 24	4))
	(assert	(gender Danie)	L M))
	(assert Jane))	(friends Danie	el Simon
© University	of Auckland	www.cs.auckland.ac.nz/~ian/	ian@cs.auckland.ac.nz



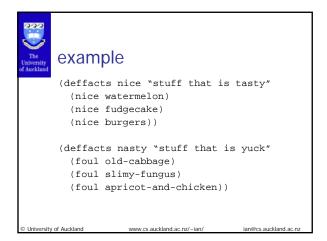


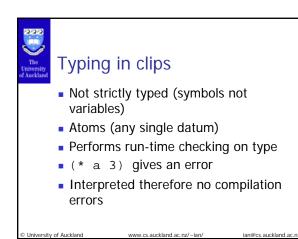




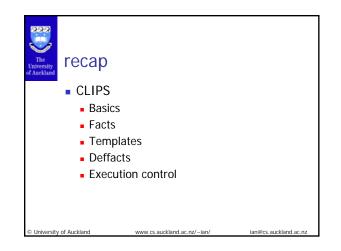
	Car Frame or	Template Example	
The University	Slots	Fillers	
of Auckland	Name	car name	
	Туре	sedan, sports, station_wagon	
	Manufacturer	GM, Ford, Chrysler, Toyota	
	Owner	Name of owner	
	Wheels	4, 6	
	Transmission	manual, automatic gasoline, diesel, methanol	
	Engine		
Condition		lemon, OK, peach	
	Under-warranty	no, yes	
© University	of Auckland www.cs.a	uckland.ac.nz/~ian/ ian@cs.auckland.ac.nz	

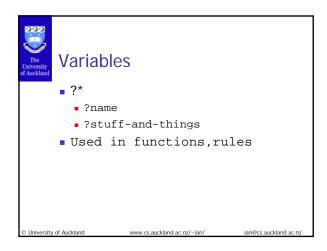
The University of Auckland	Car Insta	nce Example	
	<u>Slots</u>	Fillers	
	Name	Alice's car	
	Туре	station_wagon	
	Manufacturer	GM	
	Owner	Alice M. Agogino	
	Wheels	4	
	Transmission	manual	
	Engine	gasoline	
	Condition	OK	
	Under-warranty	yes	
© University	of Auckland w	ww.cs.auckland.ac.nz/~ian/	ian@cs.auckland.ac.nz

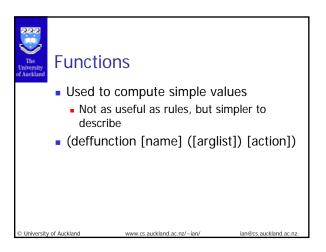




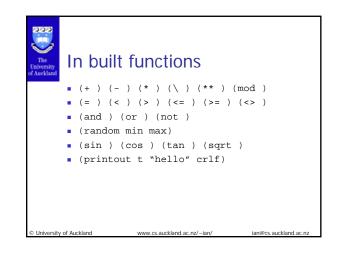


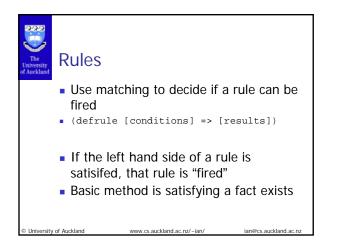


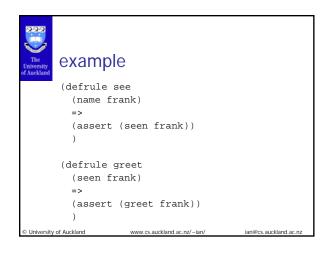


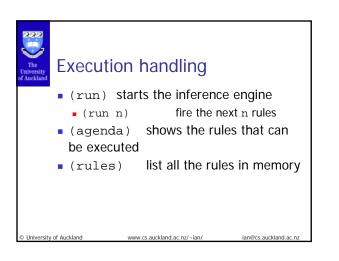


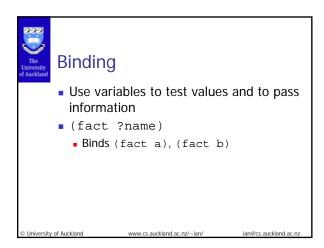
University of Auckland	le			
(deffunction fibbonacci (?f)				
(if (or	(= ?f 1) (= ?f 0))then			
1				
else				
(+	(fibbonacci (- ?f 1))			
	(fibbonacci (- ?f 2))			
)			
)				
)				
© University of Auckland	www.cs.auckland.ac.nz/~ian/	ian@cs.auckland.ac.nz		

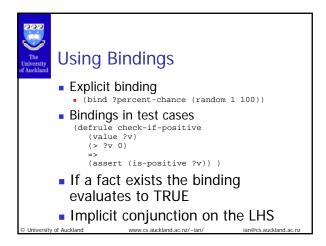


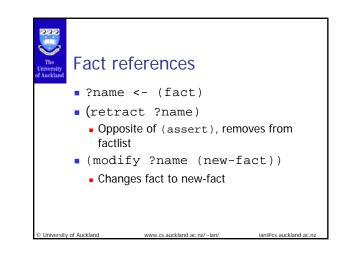


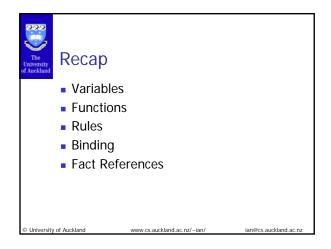


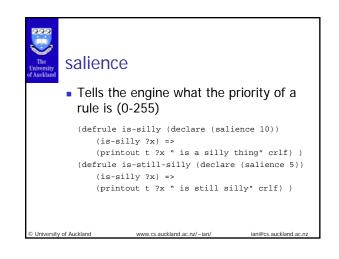


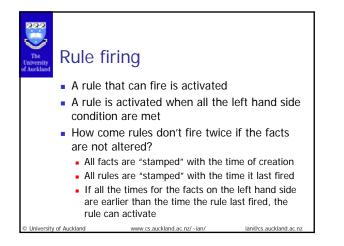














www.cs.auckland.ac.nz/~ian/

© University of Auckland

ian@cs.auckland.ac.



© University of Auckland

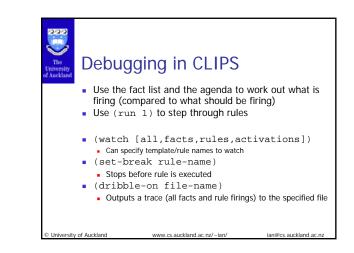
strategies

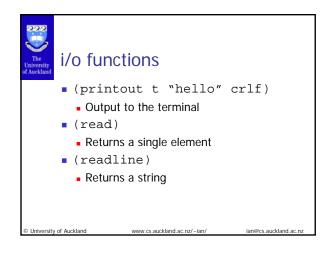
- Depth: newer activated rules first
- Breadth: newer activated rules last
- Simplicity: rules with less conditions first
- Complexity: more conditions first
- LEX: fire rules that use more recent facts
- MEA: uses the time tag of the first element

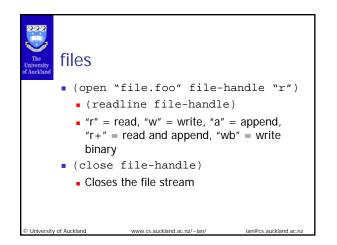
www.cs.auckland.ac.nz/~ian/

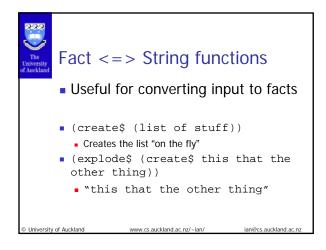
ian@cs.auckland.a

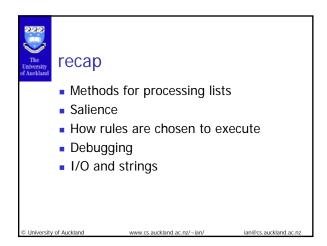
Random: select randomly

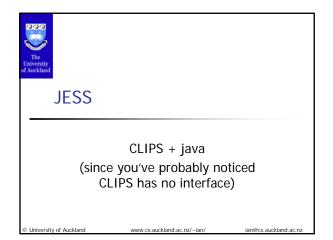


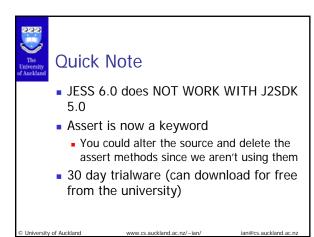


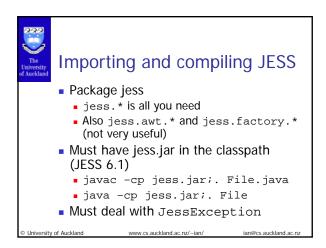


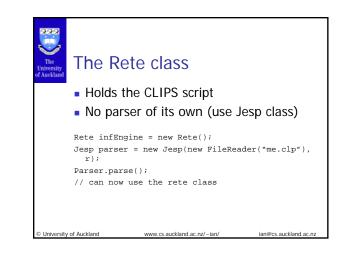


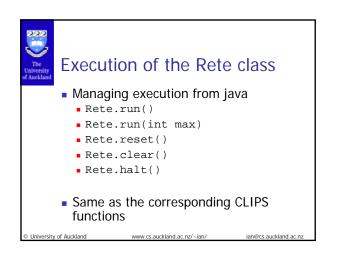


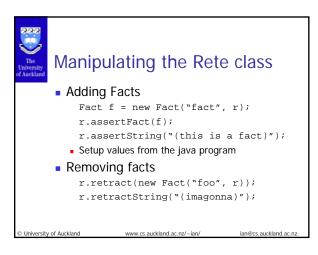


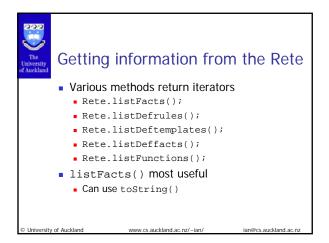


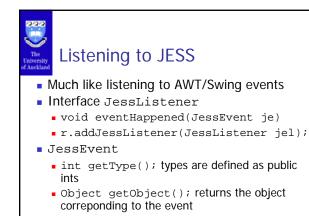


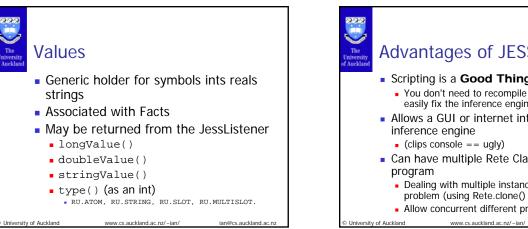


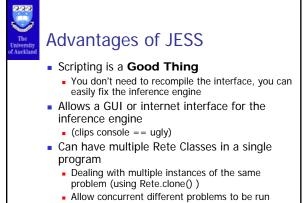






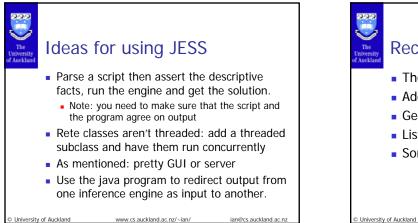






www.cs.auckland.ac.nz/~ian/

ian@cs.auckland.a



Recap

University of Auckland

- The Rete and Jesp classes
- Adding information to the Rete
- Getting information from the Rete

www.cs.auckland.ac.nz/~ian/

ian@cs.auckland.ac

- Listening to the Rete
- Some tips and ideas