2	21 Sep	**Introduction to facts and rules in Jess**
		
		1. [First steps with Jess](#). You will need to unzip [Jess71p2.zip](#) (30 day version) and use the following 3 Jess programs.
		  ● [baby.clp](#)
		  ● [rules.clp](#)
		  ● [faucet.clp](#)
		1. Goto chapter 4 of [Jess in Action](#), page 41, read through sections 4.1 and 4.2. and start trying some code fragments. You can try out those code fragments which begin with the Jess prompt `Jess>` (first one is on page 46). Note, you can copy from a PDF document and paste into the Command Prompt window.
		2. Using the Jess functions **bind** and **read** along with two local variables **?u** and **?v**, write Jess code which inputs 2 numbers, adds them and then displays the result.

3	28 Sep	**Lists and Control Structures in Jess** Try out some of the basic Jess code form section 4.2 of [Jess in Action](#), Experiment with bind?, create$, foreach, nth$, first$, rest$, length$. **Exercises - [some solutions](#)** 1. Use create$ to build a list called *?favs* of your favourite CDs, DVDs, sound tracks or pets. 2. Write a function *count-items* which count the number of items in a grocery list. Easiest way is to use *foreach*. For example: `(deffunction count-items (?list)` and so on. 3. Use *foreach* to compute the maximum value in a list of numbers. 4. Write a program with a foreach loop to calculate the average of a list of numbers. Use length$ 5. Write a program with a while loop to calculate the average of a list of numbers. Will need *length$* and *nth$*. 6. Create a modified grocery list something like (milk 5.50 butter 4.00 eggs 6.70 bread 2.30 muffins 4.55) which includes both the items and their respective costs. Then write a function *calculate-cos*t which calculates the cost of all the items in the grocery list.

		7. Look at example of how to use "if then else". Then write a function to count the number of items in a grocery list whose cost is €5 or more.

4	5 Oct	**Rules with variables**
		
		Try the following from the Jess command line interface:
		Jess>(watch all)
		Jess>(reset)
		TRUE
		Jess> (assert (man socrates))
		<Fact-1>
		Jess> (defrule mortality-rule (man ?x) => (assert (mortal ?x)))
		TRUE
		Jess> (run)
		1
		Jess> (facts)
		f-0 (MAIN::initial-fact)
		f-1 (MAIN::man socrates)
		f-2 (MAIN::mortal socrates)
		For a total of 3 facts.
		
		Recursive rules
		
		Here's a more complicated example of a recursive rule involving logical and/or. Put the code in a file **ancestor.clp** and then run it.
		(defrule ancestor-rule
		  (or (parent ?a ?b)
		    (and (parent ?a ?c) (ancestor ?c ?b)))
		  =>
		  (assert (ancestor ?a ?b)))
		
		(assert (parent andy betty))
		(assert (parent betty charlie))
		(assert (parent charlie donna))
		
		Before running this, write out what you think the results should be.
		
		Unordered Data (templates and slots) - very important
		
		Make sure you are comfortable with *foreach* and *while* loops in Jess. Look at two ways to calculate the average in avg.clp. Skip chapter 5

and go straight to chapter 6 for the moment. Look at code in person.clp. Add some more person records or facts.

1. Using a rule and a function write code to count the number of males. Incrementing a global variable on the action side (RHS) of a rule might be useful.
2. Write code to find the average male age. Same approach as in (1).
3. Modify code from (2) so that it can handle incomplete data, e.g. a person whose age is unknown. E.g.
4. `(assert (person (name "Billy Bob") (gender Male)))`

To do this you could use a default value of FALSE for the age slot, see page 83 of Jess in Action. Then in the action side of the rule, put an if function which if age is false ignores it and not false adds it to sum and increments count. Note (if 3 ...) is the same as (if TRUE ...).

1. Can you write code to find the name and age of the youngest person in the database using global variables?

Solutions 1, 2, 3, 4 (4 solutions to ex 4)

5	12 Oct	

Lab Marking this week for following exercises

The following two related exercises are to be completed and checked in the lab. You can use code from last week problem as a guide if you wish.

1. Using an (if ...) in the RHS (after =>) of the rule, extend the the Jess program oldest1.clp by completing the rule and function so that the name and age of the oldest male is displayed.
2. Modify oldest1.clp so that it uses a constraint with a predicate function on the LHS of the rule rather than an *if* function on the RHS. See page 106 of Jess in Action. Save program as oldest2.clp.