



## Knowledge Based System Course

Lab Objective:	<ul style="list-style-type: none"> <li>Continue explaining CLIPS basic commands by example</li> </ul>
Topics:	<ul style="list-style-type: none"> <li>Facts <ul style="list-style-type: none"> <li>Multifield Slots</li> <li>Slots With Default Values</li> </ul> </li> <li>Commands For Manipulating Constructs <ul style="list-style-type: none"> <li>Modify Facts</li> </ul> </li> <li>Rules And Matching <ul style="list-style-type: none"> <li>Multislot Vairable</li> <li>Single Field &amp; Multifield Wildcards</li> <li>Field Constraints (Not, Or, And)</li> <li>Saving fact-index</li> <li>Bind Function</li> <li>Test Function</li> </ul> </li> </ul>
Contents	<p>1- ////allowed-values :if slot not entered default is 1st value</p> <p>2- ////NUMBER slots if not entered default is 0</p> <p><b>3- Multifield Slots, Slots With Default Values:</b></p> <p>Define a template for a person (name, age, gender)</p> <pre> (deftemplate person (slot age (type INTEGER) (default 18)) (multislot name (type STRING)) (slot gender (allowed-values f m)) ; f=female and m=male (slot degree (type SYMBOL) (allowed-symbols none bsc msc phd)) ) </pre>



```
clips>(assert (person (name "Ali" "Ahmed") (gender m)))  
What is the value for field age?
```

#### **4- Multislot Variables:**

```
CLIPS> (deffacts people  
  (person (name "Ali") (age 21) (gender m) (degree bsc))  
  (person (name "Mona" "Ali") )  
  (person (name "Ahmed" "Saleh") (age 14) (gender m) (degree  
    msc)))  
CLIPS>(reset)
```

```
CLIPS> (defrule army  
  (person (gender m) (name ?name))  
  =>  
  (printout t ?name " is eligible for army" crlf))  
CLIPS> (run)  
Ali is eligible for army
```

what's about "Ahmed Saleh" (multislot)?

```
CLIPS> (defrule army  
  (person (gender m) (name $?name))  
  =>  
  (printout t ?name " is eligible for army" crlf))  
  
CLIPS> (run)  
("Ahmed" "Saleh") is eligible for army  
("Ali")is eligible for army
```

#### **5- Single Field & Multifield Wildcards:**

To match any value, if we don't need that value later

```
CLIPS> (defrule army
```



```
(person (gender m) (name ?firstname ?))
=>
(printout t ?firstname " is eligible for army" crlf))
```

But this will print only people with 2 names inserted so "Ali" will not be printed. What should we do?  
 Change the condition from *(person (gender m) (name ?firstname ?))* to *(person (gender m) (name ?firstname \$?))*  
 (\$?) Matches 0 or more but (?) matches only 1.

```
CLIPS> (defrule army
  (person (gender m) (name $?))
  =>
  (printout t "someone is eligible for army" crlf))
```

#### **6- Field Constraints (Not, Or, And):**

```
(defrule army
  (person (gender m) (name $?name) (degree bsc/msc/phd))
  =>
  (printout t ?name " is eligible for army officer" crlf))
```

OR

```
(defrule army
  (person (gender m) (name $?name) (degree ~none))
  =>
  (printout t ?name " is eligible for army officer" crlf))
```

To assign the value to a variable

```
(defrule army
  (person (gender m) (name $?name) (degree
?degree&bsc/msc/phd))
```



=>

*(printout t ?name " is eligible for army officer as he holds a "  
?degree "degree" crlf))*

### **7- Test function in constraints:**

*CLIPS>(defrule army  
(person (gender m) (name \$?name) (age ?age))  
(test (> ?age 18))*

=>

*(printout t ?name " is eligible for army" crlf))*

Attaching a predicate-expression(true or false expresssion)  
directly to the slot (another way).

*CLIPS> (defrule army  
(person (gender m) (name \$?name) (age ?age&:(> ?age 18)))  
=>  
(printout t ?name "is eligible for army" crlf))*

### **8- And, Or and Not can be used for facts too:**

Example:

*(deftemplate triangle (slot side1)(slot side2)(slot side3))  
(deffacts triangles  
(triangle (side1 3)(side2 4)(side3 3)))*

*(defrule isosceles\_tri  
(or (triangle (side1 ?s)(side2 ?s))  
(triangle (side1 ?s)(side3 ?s))  
(triangle (side2 ?s)(side3 ?s))  
)*

=>

*(printout t "There is an isosceles triangle" crlf))*



### **9- Saving fact-index and bind function:**

Suppose you have a rule that uses a fact to fire (in the pattern part) and once it fires you want to retract this fact or modify its content (in the action part), then you should save the fact index of this fact because both modify and retract take fact-index as a parameter).

#### Example:

```
(deffacts f (ctr 0))  
(defrule rul  
  ?num <- (ctr ?c)  
  (test (neq ?c 5))  
=>  
  (bind ?new-c (+ ?c 1))  
  (printout t ?c crlf)  
  (retract ?num)  
  (assert (ctr ?new-c)))
```

Note that bind function in general can only be used on the RHS of the rule. It binds a NEW variable to a value. It should not bind an existing variable with a value to a new value because in clips variables only take one value.

**10- Modify Function:** In the previous example we used retract and assert to change the value of a fact, can't we just modify directly in the fact? We can!!  
Modify function modifies slot values of a fact taking as parameter the fact index and the slot name and new value  
(modify <fact-index> (<slot-name> <slot-value>)+)

The fact in the previous example was an ordered fact, so we should change it to an unordered one using a deftemplate:

Let them try to do it themselves.

Solution:

```
(deftemplate ctr (slot val))
```



```
(deffacts f
  (ctr (val 0)))
(defrule rul
  ?num <- (ctr (val ?c))
  (test (neq ?c 5))
  =>
  (bind ?new-c (+ ?c 1))
  (printout t ?c crlf)
  (modify ?num (val ?new-c))
)
```

- 11-** Save command to the cursor commands you just wrote:  
(save "myfile.clp")