Daniel Leeds, 15-212 R03, September 12, 2007

Syntax:

```
local
   val a = …
   fun f z = …
   val k = …
in
   fun foo n m = f (a+k * (m-n))
end;

foo 3 4;
```

Functions and variables bound between the keywords “local” and “in” are visible only to the expressions evaluated between “in” and “end.” The values bound between “in” and “end” are accessible to expressions outside the local - in - end framework. Similar syntax applies for let - in - end

Lists:

Given list L,

hd L	returns first *element* in the list
tl L	return *list* containing everything but the first element

Merge Sort:

```
fun merge p ([], ys) = ys
 |   merge p (xs,[])  =  xs
 |   merge p (x::xs,y::ys) =
             if x::(merge p (xs,y::ys))
             else if p(y,x)
                  then y::(merge p (x::xs,ys))
                  else x::y::(merge p (xs,ys))
```

```
fun split [] = ([],[])
 |    split [x]=([x],[])
 |    split (x::y::L) =
            let
               val (L1,L2) = split L
            in
               (x::L1,y::L2)
            end
```

```
fun sort p [] = []
 |    sort p [x] = [x]
 |    sort p L =
            let
                val (L1,L2) = split L
            in
                merge p (sort p L1, sort p L2)
            end
```