# Applicative programming: basic tools

Umut Özge

COGS 502: Symbols and Programming
METU, Informatics

# Global variables

```
(defparameter *grades*
  '((e842222 86) (e850421 98) (e790059 79) (e170139 45)
    (e917272 0)  (e989199 75) (e877076 96) (e511096 83)
    (e386463 91) (e337777 90) (e861067 54) (e801835 70)
    (e493198 85) (e352336 82) (e243952 91) (e595538 47)
    (e304901 0)  (e548145 70)))
```

# MAPCAR

# LAMBDA

```
(defun cube (x)
  (expt x 3))
```

# LAMBDA

```
(defun cube (x)
  (expt x 3))
```

```
(lambda (x) (expt x 3))
```

# LAMBDA

```
(defun cube (x)
  (expt x 3))
```

```
(lambda (x) (expt x 3))
```

```
(mapcar
  #'(lambda (x)
      (cons (car x)
            (list (if (>= (cadr x) 60) 'passed 'failed))))
  *grades*)
```

# REDUCE

```
(reduce #'+ '(1 2 3 4 5))
```

# REDUCE

```
(reduce #'+ '(1 2 3 4 5))
```

```
(reduce #'+ (mapcar #'cadr *grades*))
```

# REDUCE

```
(reduce #'+ '(1 2 3 4 5))
```

```
(reduce #'+ (mapcar #'cadr *grades*))
```

```
(reduce
     #'(lambda (x y) (if (> x y) x y))
     (mapcar #'cadr *grades*))
```

# REMOVE-IF

```
(remove-if #'zerop (mapcar #'cadr *grades*))
```

# MAKE-LIST