CLOS

Classes

```
(defclass person (standard-object)
    ((name :accessor person-name
           :initarg :name
           :allocation :instance)
     (address :accessor person-address
              :initarg :address
              :allocation :instance))
(:documentation "This is a class for person objects."))
```

Class and Superclasses

```
(defclass person (standard-object)
    ((name :accessor person-name
           :initarg :name
           :allocation :instance)
     (address :accessor person-address
              :initarg :address
              :allocation :instance))
(:documentation "This is a class for person objects."))
```

Slots and Slot Options

```
(defclass person (standard-object)
    ((name :accessor person-name
           :initarg :name
           :allocation :instance)
     (address :accessor person-address
              :initarg :address
              :allocation :instance))
(:documentation "This is a class for person objects."))
```

Class Options

```
(defclass person (standard-object)
    ((name :accessor person-name
           :initarg :name
           :allocation :instance)
     (address :accessor person-address
              :initarg :address
              :allocation :instance))
(:documentation "This is a class for person objects."))
```

Instances & Accessors

- (defparameter *dilbert*
    (make-instance ‘person :name “Dilbert” :address “Brussels”))

- (person-name *dilbert*)
  => “Dilbert”

Generic Functions & Methods

- (defgeneric display (object))

- (defmethod display ((object person))
    (print (person-name object))
    (print (person-address object)))

- (display *dilbert*)

Inheritance

- (defclass employee (person)
    ((employer :accessor person-employer)))

- (defmethod display ((object employee))
    (call-next-method)
    (print (person-employer object)))

Classes & Objects

- Multiple inheritance.
- Classes can be dynamically redefined.
- Objects can dynamically change their class.
- Objects can be reinitialized.

Classes & Objects

- Standard slot options:
    - :initform, :initarg
    - :allocation, :type
    - :reader, :writer, :accessor
    - :documentation

Classes & Objects

- Standard class options:
  - :default-initargs
  - :documentation
  - :metaclass

Generic Functions & Methods

- (defgeneric display (object)
    (:documentation “Displays objects.”)
    (:method ((object standard-object))
      (print “An object.”)))

Generic Functions & Methods

- Standard generic function options:
  - :argument-precedence-order
  - declare
  - :documentation
  - :method-combination, :method-class
  - :generic-function-class

Generic Functions & Methods

- (defmethod display ((object person))
  ...)

- (defmethod display :before ((object person))
  ...)

- Standard method combination allows for primary, :before, :after and :around methods.

Generic Functions & Methods

- (defgeneric display (object)
    (:method-combination progn :most-specific-last))

- (defmethod display progn ((object person))
    (print (person-name object))
    (print (person-address object)))

- (defmethod display progn ((object employee))
    (print (person-employer object)))

Generic Functions & Methods

- Use define-method-combination to define new method combinations.