Object-Oriented Programming: Inheritance

5 Dec 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

Reminders:

- ***journals** in folder*
- ***quiz** ch12 today*

Quiz ch12: 4 questions, 20 marks, 10 min

- Define "endianness" in your own words
- List advantages and disadvantages of linked lists vs. M2 arrays
- Declare a circular, doubly-linked (bidirectional) list that stores CARDINALs, and initialize it with a single entry
- Draw a diagram illustrating a circular, doubly-linked list with three entries, storing CARDINALs 0, 1, 2. Include all relevant fields and pointers, and indicate any NIL pointers

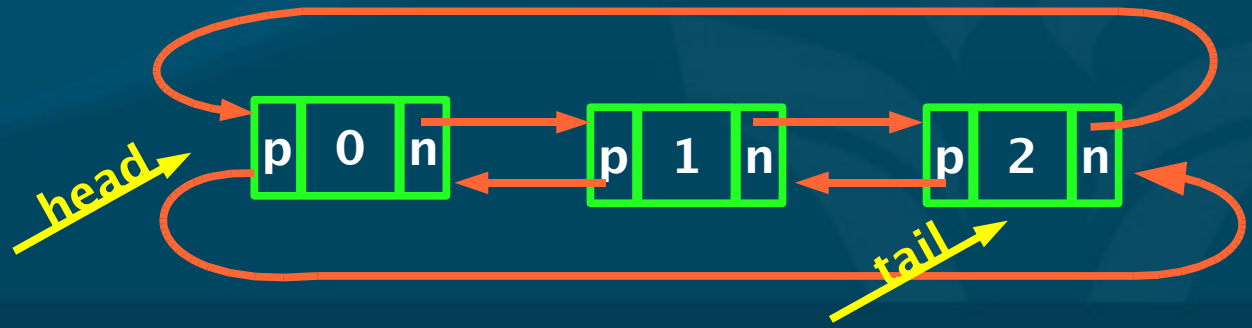
Quiz ch12 answers: #1-2

- Define "endianness" in your own words
  - Ordering of bytes within a word
    - e.g., "1A 2B 3C 4D" big-endian = "4D 3C 2B 1A" little-endian
- List advantages and disadvantages of linked lists vs. M2 arrays
  - Linked-lists: dynamic, resizable, insert near beginning is faster
  - Arrays: built-in, easier to use, no worries about memory leaks or dereferencing NIL pointers

Quiz ch12 answers: #3-4

```
TYPE
    CDList = POINTER TO CDListNode;
    CDListNode = RECORD
        data : CARDINAL;
        prev : CDList;
        next : CDList;
    END;
VAR
    myListHead, myListTail : CDList;
```

```
BEGIN
    NEW (myListHead);
    myListHead^ :=
      CDListNode {0,
      myListHead,
      myListHead};
    myListTail := myListHead;
```



Review of last time

- Object-oriented programming:
  - Procedural vs. OO
  - Objects and messages
  - Methods and attributes
    - Class interface
  - Classes and instances

Example of OO terminology

- Tell our dog Fido to fetch a stick:

```
fido.Fetch (theStick);
```

  - Dog is the class (type of object)
  - fido is the instance (variable of given type)
  - Fetch() is a method (procedure)
    - theStick is a parameter to the method
    - theStick is itself an object
- Dogs may also have attributes: color, owner, etc.

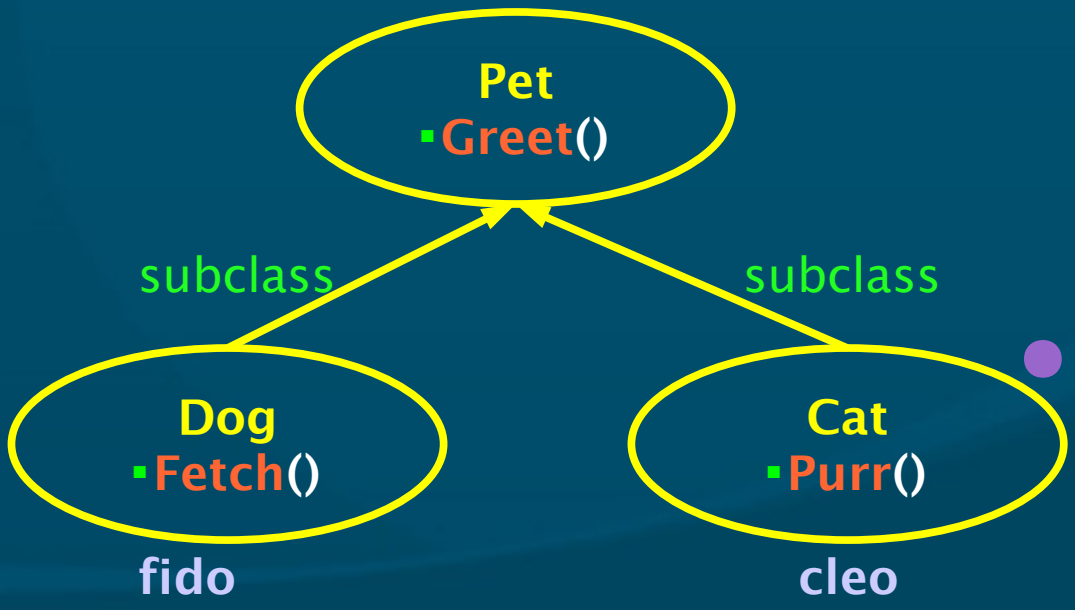
```
fido.color := brown;
```

Constructors and factories

- An object factory is an abstract concept describing the process of creating a new object
  - In M2: NEW()
  - Classes have factories to churn out new instances: create and initialize
- Constructors are functions run automatically upon creation of a new object
- e.g. Doubly-linked list example
- e.g. Default color of new Dog

Inheritance

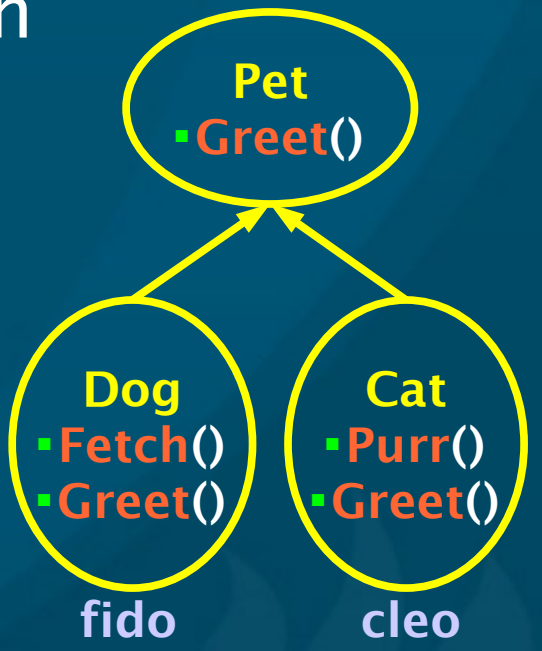
- Classes (object types) may also be derived from other classes
  - Subclasses inherit everything from parents
  - May also add their own methods/attributes
    - Both fido and cleo can Greet(), but cleo won't Fetch()



- Subclass vs. instance:
  - We say, "fido is a Dog", but
  - "A Dog is a kind of Pet"

Overriding and virtual functions

- A subclass can override a method in the parent class by redefining it
  - Parent's version is hidden
  - Parent is also called superclass
- In fact, the parent need not even have a body for the method:
  - Virtual function (or method)
  - Just declares the name and how to invoke
- Polymorphism: all Pets can Greet(), but Dogs Greet() differently from Cats



Summary of OO

- ADT-oriented rather than action-oriented
  - Everything is an object
- Encapsulate everything you need to use an ADT within its object class definition
- Action happens by passing messages to objects
  - Objects define interfaces: how to use
- Classes can inherit from other classes
  - And override and/or add to inherited stuff
- Keywords: object, procedural vs. OO, message, method, interface, attribute, instance, factory, class, inheriting, overriding, virtual function

TODO items

- Lab 10 due today/Tues/Wed: §11.10 #(25 / 30)
- Paper due on Wed!
- Final exam: Wed 14Dec 2–4pm here