



ECE12: Introduction to Programming

Lecture 17

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 17: Overview

- Object-oriented Programming
 - Putting it all together
 - Classes vs. objects
 - References and namespaces
 - Special methods and overloading
 - Class composition
 - Containment relationship
 - Object references as members of classes

Object-Oriented Programming

- Example: `class Time` (file `time3.py`)

```
class Time:
    """abstract data type for representation of time"""

    def __init__(self, hour=0, minute=0, second=0):
        """creates a time object and initializes it"""
        self.SetTime(hour, minute, second)

    def __del__(self):
        """cleans up and destroys a time object"""
        # nothing to do here, but let's print a message
        print "Time object being destroyed!"

    def __str__(self):
        """returns an informal string representation"""
        return "%02d:%02d:%02d" % \
            (self.__hour, self.__minute, self.__second)

    def __repr__(self):
        """returns a formal string representation"""
        return "Time(%d,%d,%d)" % \
            (self.__hour, self.__minute, self.__second)

    ...
```

Object-Oriented Programming

- Example: `class Time` (file `time3.py`)

```
...
    def __add__(self, other):
        """overloads the + operation for time objects"""
        return self.Add(other)
```

- Interactive Example:

```
% python
>>> from time3 import Time
>>> t1 = Time(10, 20)
>>> t2 = Time( 2, 11, 30)
>>> print t1, t2
10:20:00 02:11:30
>>> t3 = t1 + t2
>>> print t3
12:31:30
>>> del t1
Time object 10:20:00 being destroyed!
>>> ^d
Time object 12:31:30 being destroyed!
Time object 02:11:30 being destroyed!
%
```

Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

Global Namespace

...

Data memory:

Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

Global Namespace

...

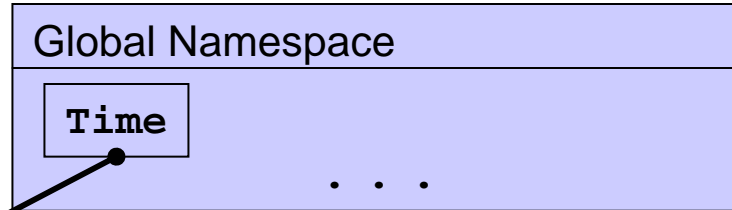
1. compile file "time3.py"

Data memory:

Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```



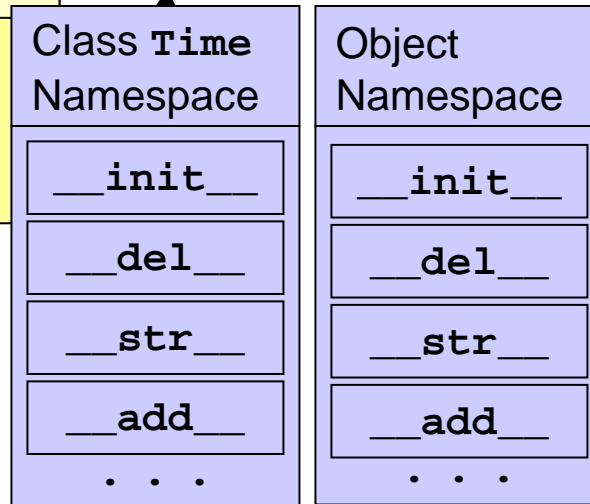
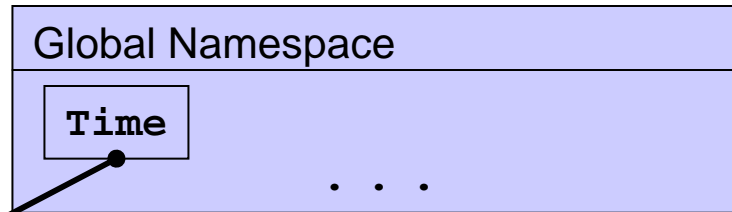
1. compile file "time3.py"
2. create reference to class Time



Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```



1. tmp = new object of type Time

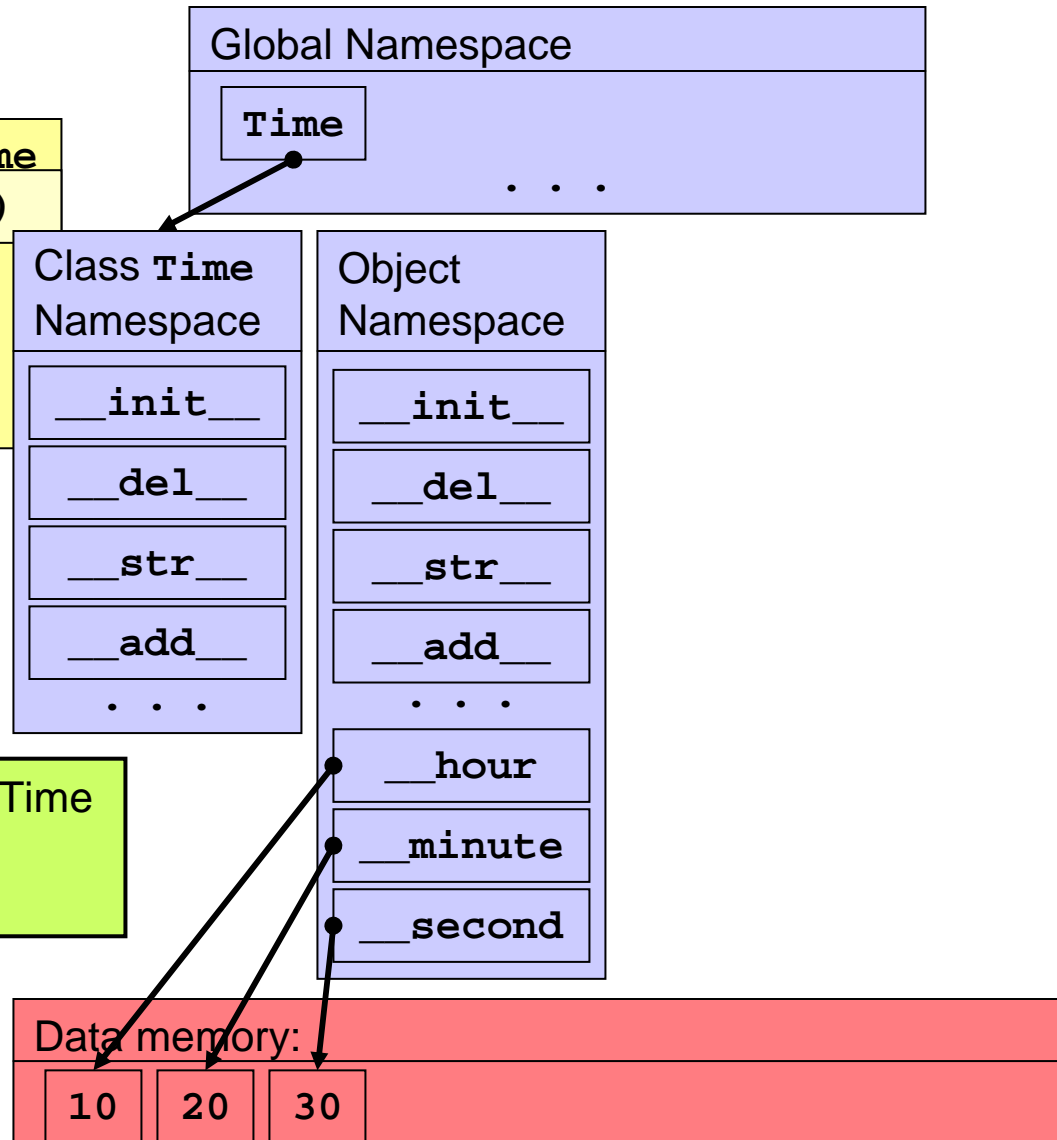
Data memory:

Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

1. tmp = new object of type Time
2. __init__(tmp, 10, 20, 30)

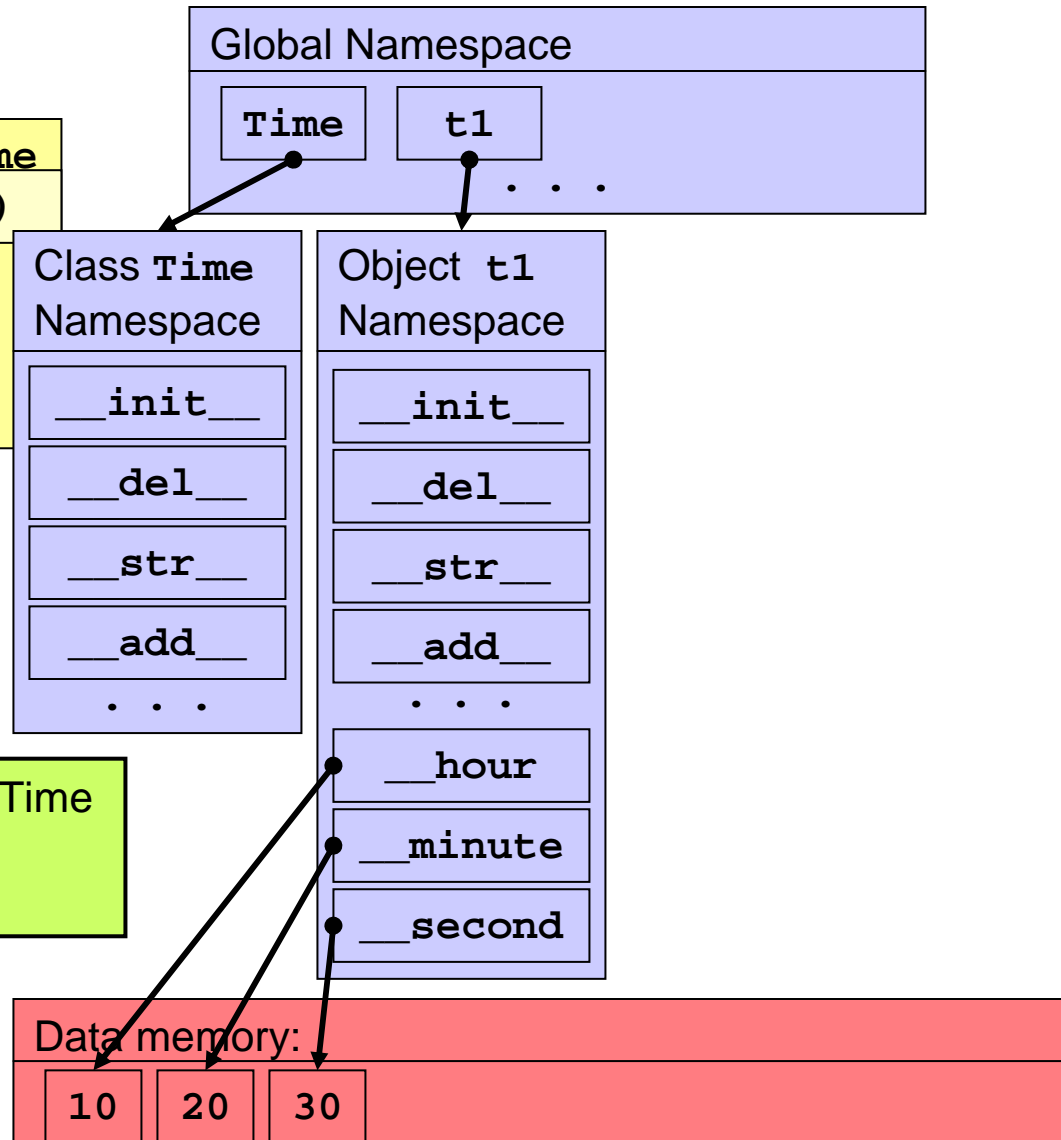


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

1. tmp = new object of type Time
2. __init__(tmp, 10, 20, 30)
3. t1 = tmp

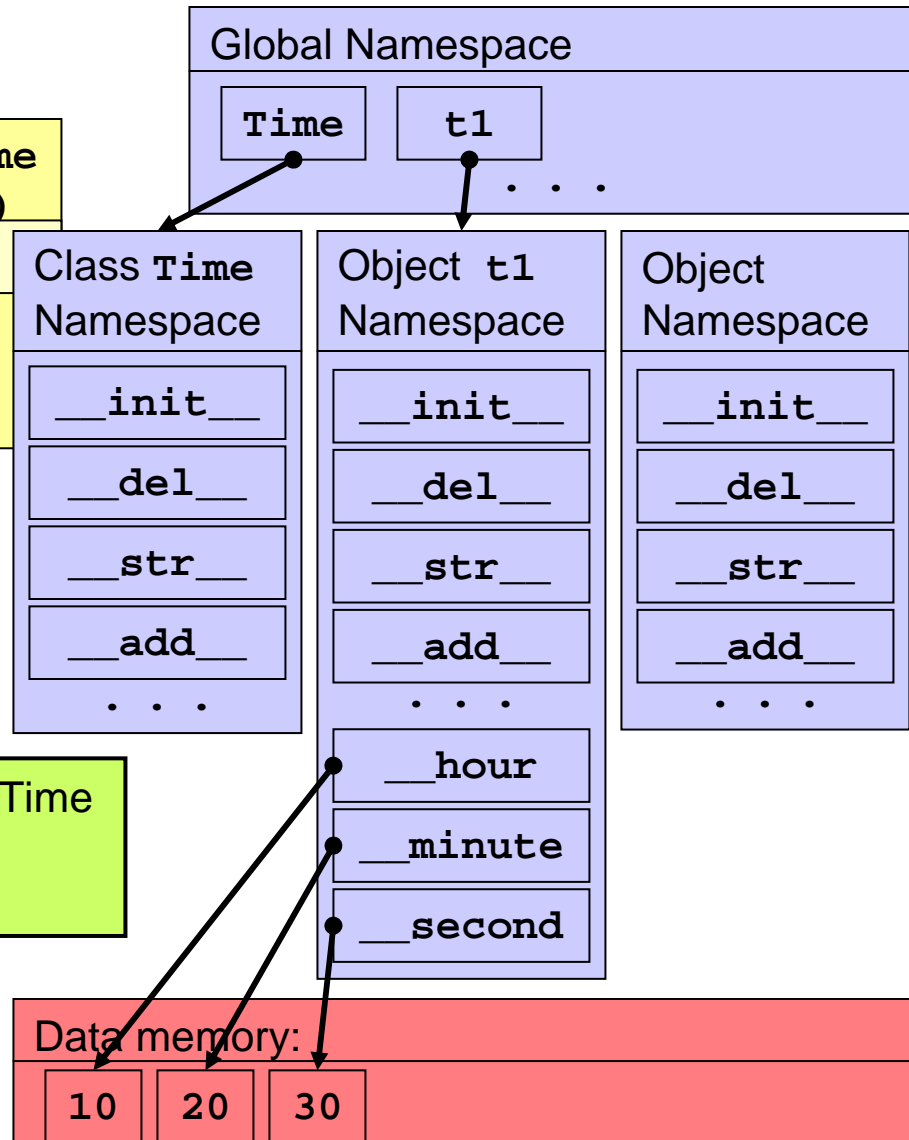


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

1. tmp = new object of type Time

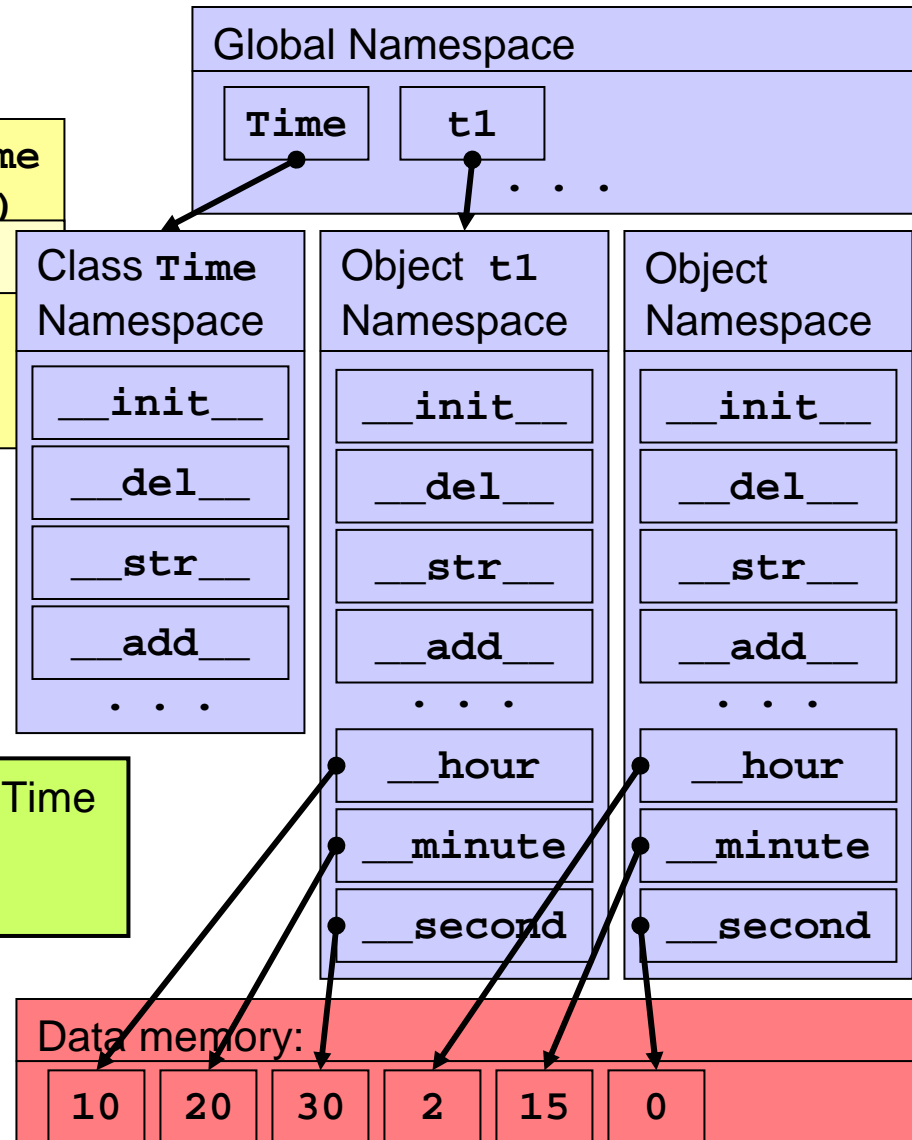


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

```
1. tmp = new object of type Time
2. __init__(tmp, 2, 15, 0)
```

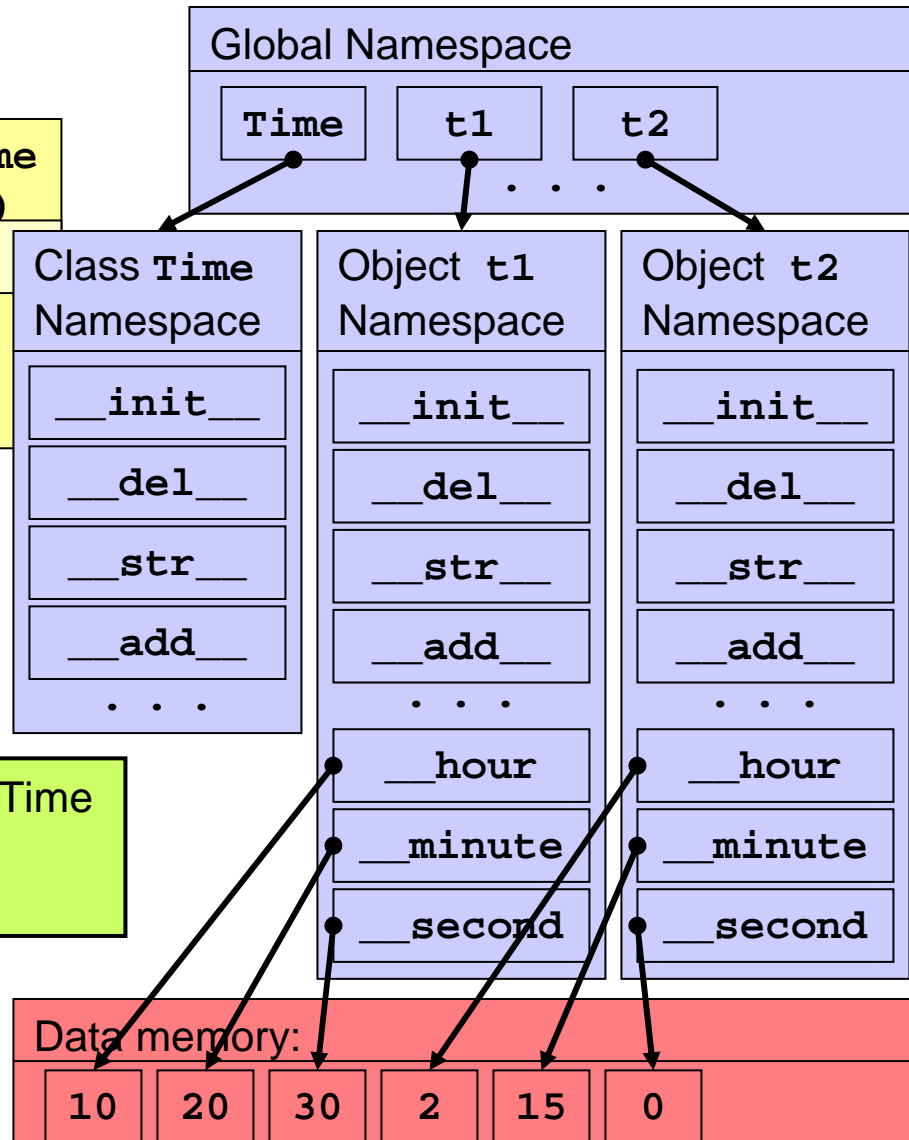


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

```
1. tmp = new object of type Time
2. __init__(tmp, 2, 15, 0)
3. t2 = tmp
```

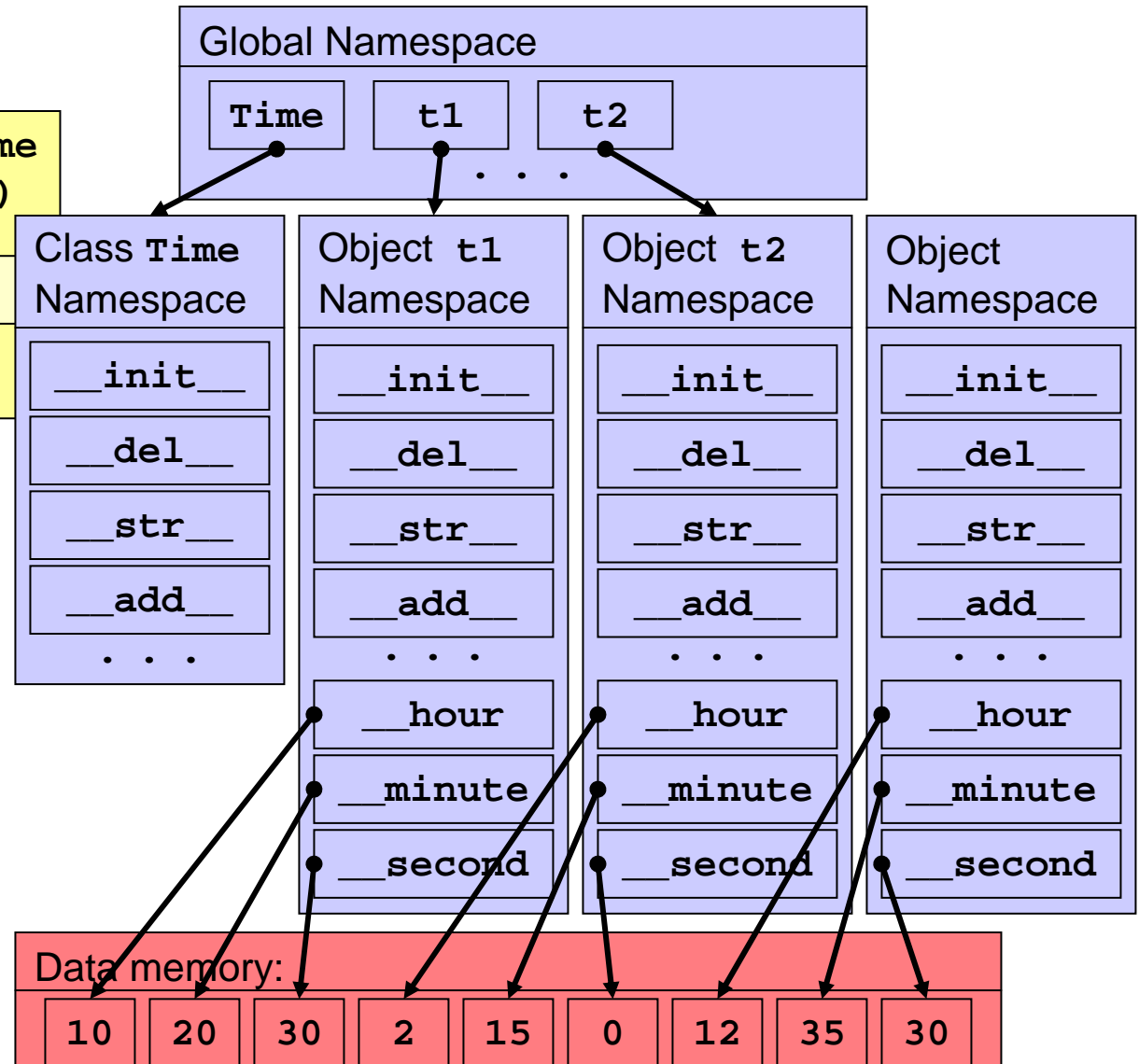


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

```
1. tmp = __add__(t1, t2)
```

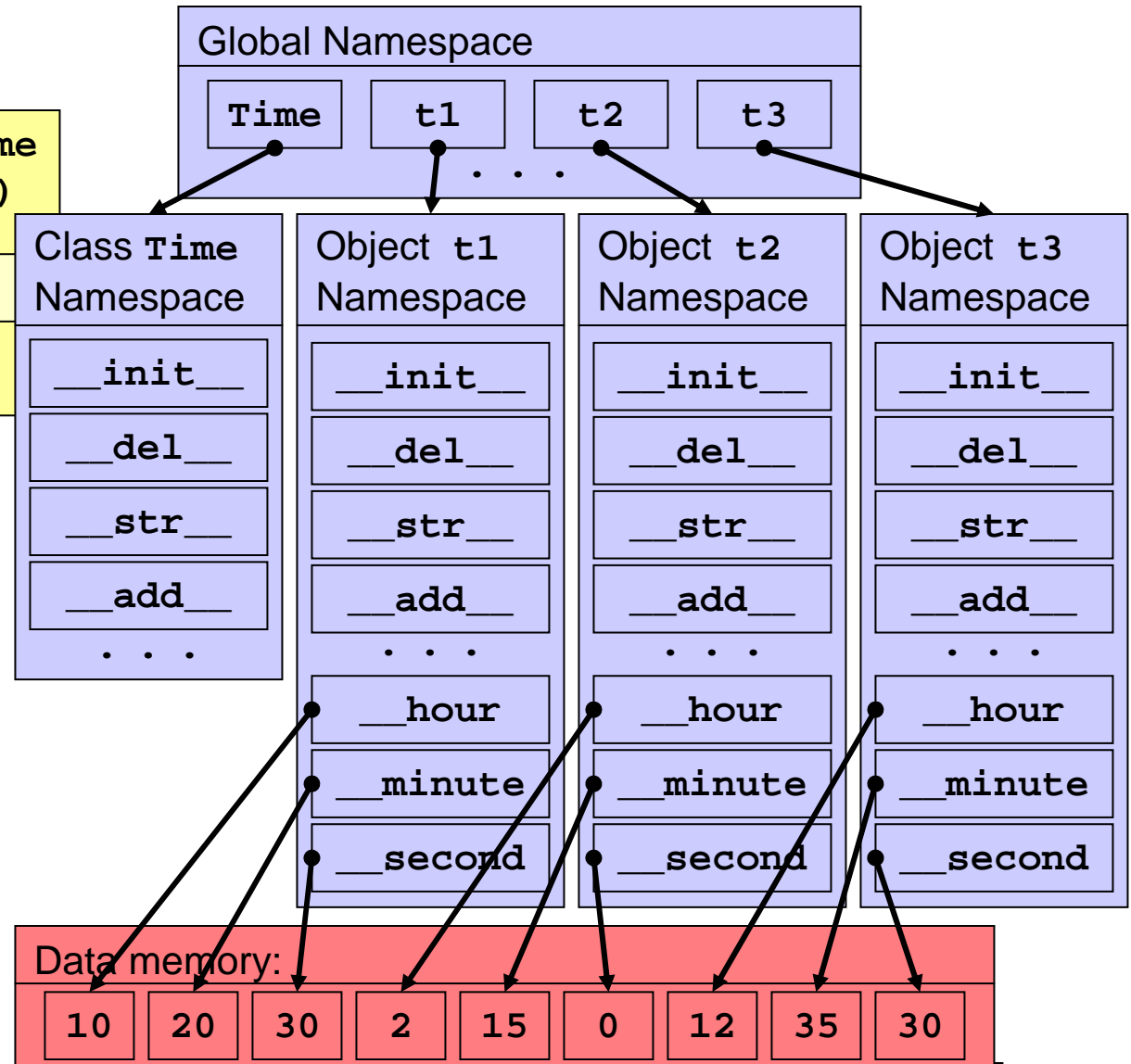


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

```
1. tmp = __add__(t1, t2)
2. t3 = tmp
```

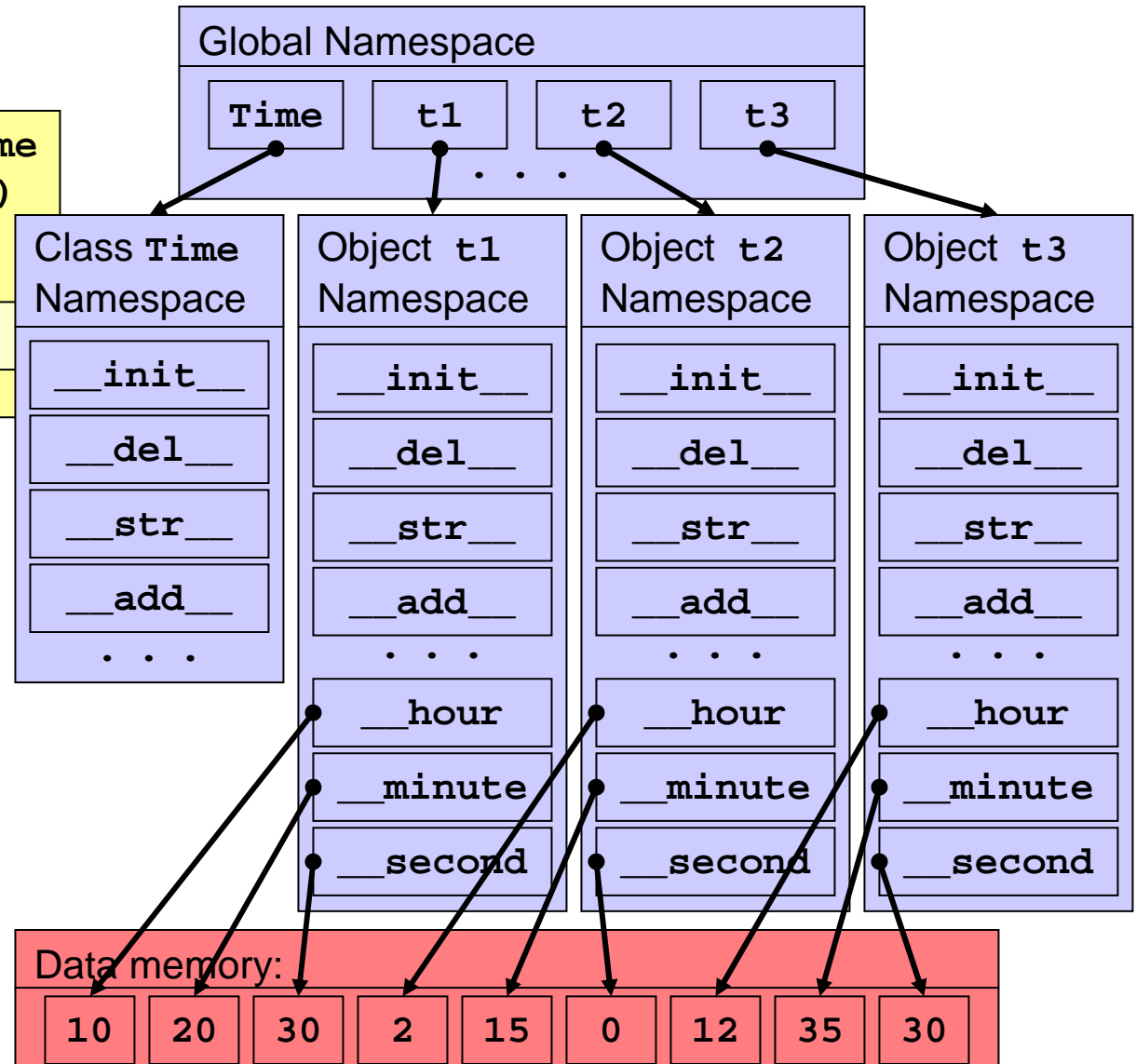


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

```
1. tmp = __str__(t3)
2. print tmp
```



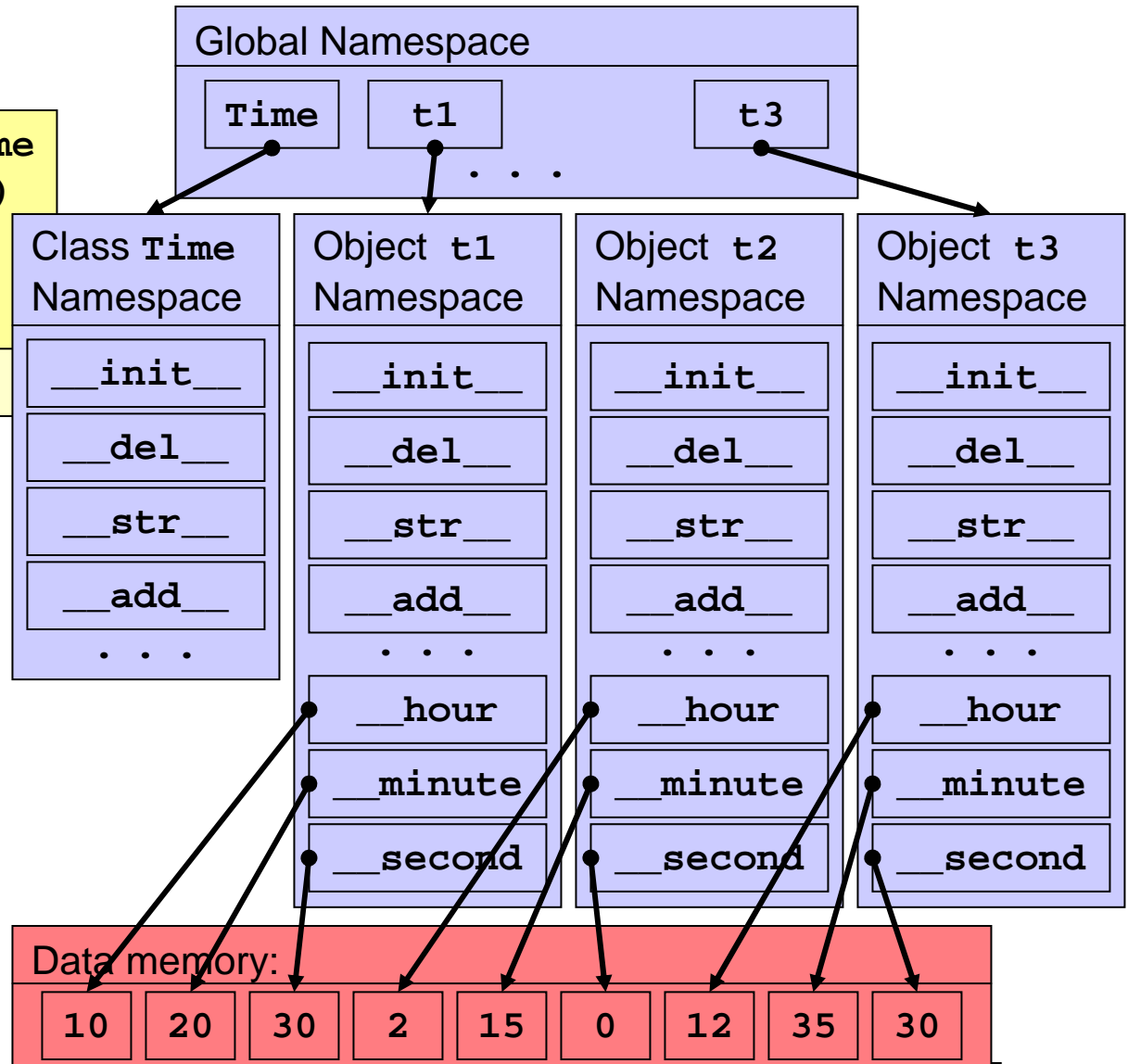
Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
```

```
del t2
```

```
1. __del__(t2)
```

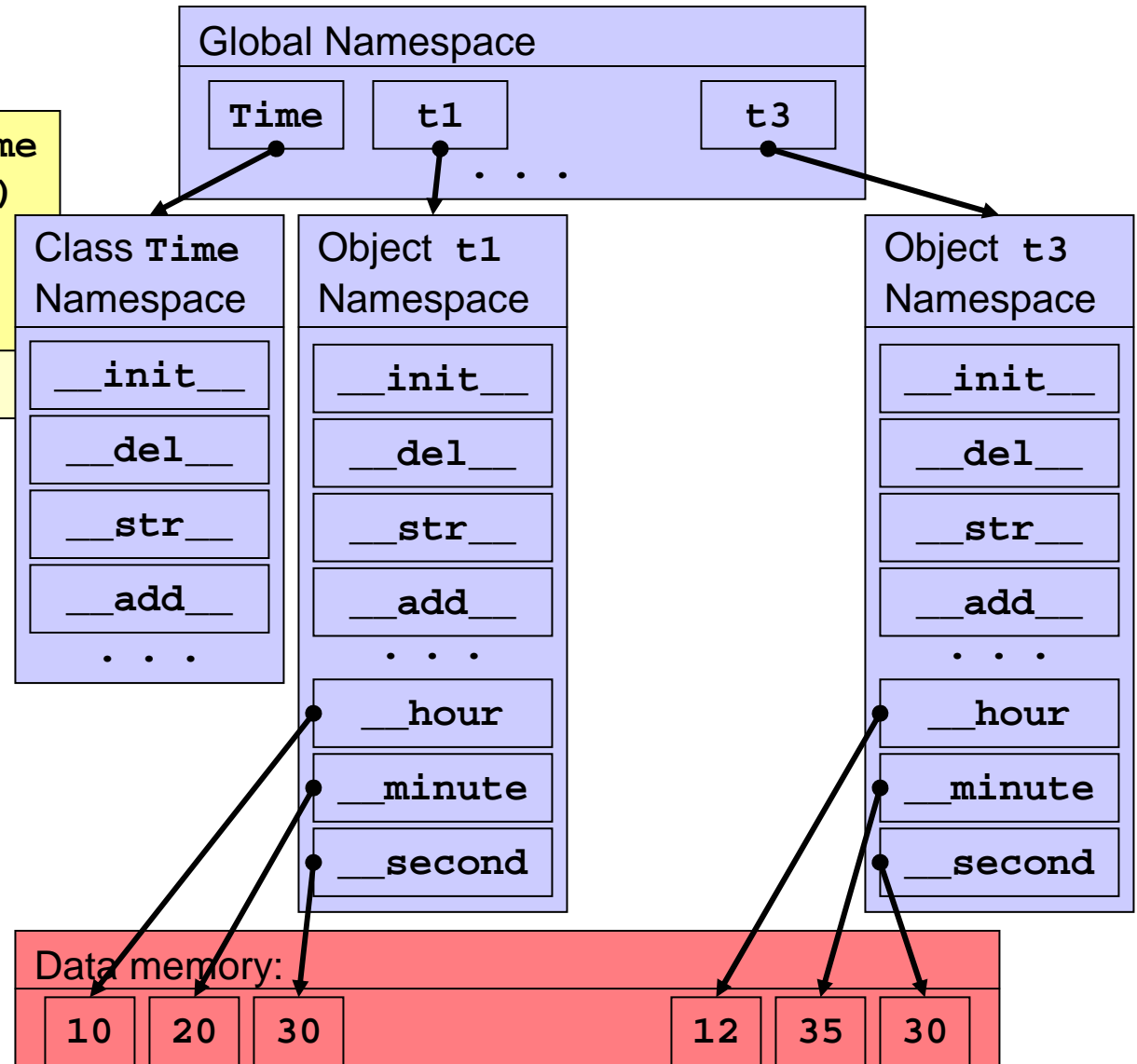


Object-Oriented Programming

- Example:

```
from time3 import Time
t1 = Time(10, 20, 30)
t2 = Time( 2, 15)
t3 = t1 + t2
print t3
del t2
```

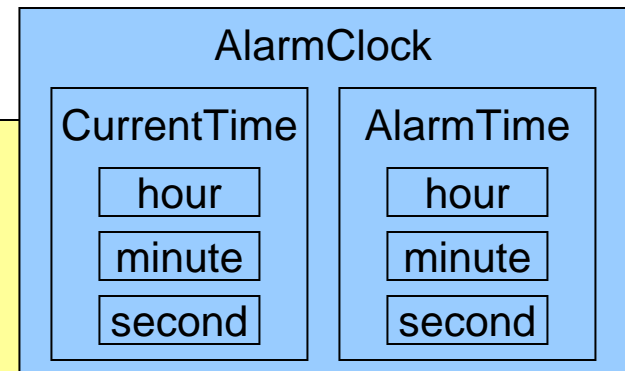
1. `__del__(t2)`
2. garbage collection



Object-Oriented Programming

- Class Composition
 - Classes can be hierarchically organized in a containment relationship (membership)
 - Object references can be used as members
 - A class contains objects of other classes as members
 - Example: Alarm clock

```
class Time:
    def __init__(self, hour, minute, second)
        ...
class AlarmClock:
    def __init__(self, CurrentTime, AlarmTime)
        ...
```



Object-Oriented Programming

- Class Composition
 - Example: Employee
 - class Employee:
 - String: FirstName
 - String: LastName
 - Integer: ID
 - Date: BirthDate
 - » Integer: Year
 - » Integer: Month
 - » Integer: Day
 - Date: HireDate
 - » Integer: Year
 - » Integer: Month
 - » Integer: Day
 - Floating point: Salary

