ECE12: Introduction to Programming
Lecture 15

Rainer Dömer

doemer@uci.edu

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

Lecture 15: Overview

- Object-oriented Programming
  - Access to object attributes
    - Direct access
    - Example: **`time1.py`**
  - Access control
    - Data integrity
    - Access methods
      - Set methods
      - Get methods
    - Chained comparisons
    - Raising exceptions
    - Example: **`time2.py`**

Object-Oriented Programming

- Example: **`class Time`**
  - Program **`time1.py`** (part 1/2)

```
# time1.py: abstract data type for representation of time
#           (version 1)
# author: Rainer Doemer
# 02/17/04 RD    initial version (similar to figure 7.1)

# class definition
class Time:
    """abstract data type for representation of time"""

    def __init__(self):           # constructor
        """creates a time object initialized to 12am"""
        self.hour   = 0  # 0-23   # data members
        self.minute = 0  # 0-59
        self.second = 0  # 0-59

    def Print(self):              # method
        """prints the value of a time object"""
        print "%02d:%02d:%02d" %  \
                (self.hour,self.minute,self.second)
...
```

Object-Oriented Programming

- Example: **`class Time`**
  - Program **`time1.py`** (part 2/2)

```
...
    def PrintAMPM(self):  # method
        """prints the time in am/pm notation"""
        h = self.hour % 12
        if h == 0:
            h = 12
        if self.hour < 12:
            ampm = "am"
        else:
            ampm = "pm"
        print "%2d:%02d:%02d %s" %\
                (h,self.minute,self.second,ampm)
```

Object-Oriented Programming

- Example: **`class Time`**
  - Interactive use of module **`time1.py`**

```
% python
>>> from time1 import Time
>>> t1 = Time()
>>> t1.Print()
00:00:00
>>> print t1.hour, t1.minute, t1.second
0 0 0
>>> t1.hour = 15
>>> t1.minute = 30
>>> print t1.hour, t1.minute, t1.second
15 30 0
>>> t1.Print()
15:30:00
>>> t1.PrintAMPM()
 3:30:00 pm
>>> t1.hour = 27
>>> t1.minute = 88
>>> t1.second = -1
>>> t1.Print()
27:88:-1
```

Access to Object Attributes

- Direct access
  - Dot-operator (`.`)
    - from inside the class: through object reference **`self`**
      - Example:
        - » Read access: **`print self.hour`**
        - » Write access: **`self.hour = 15`**
    - from outside the class: through object name
      - Example:
        - » Read access: **`print t1.hour`**
        - » Write access: **`t1.hour = 15`**
  - Direct access from outside the class
    - violates concept of information hiding!
    - can lead to an inconsistent state of an object!
      - Example:
        - » **`t1.minute = 88 # value out of range!`**

Access to Object Attributes

- Access control
  - Object interfaces: Get and Set methods
    - Access object data under program control
      - Get: method to obtain data from an object
        - » Read access: **`print t1.GetHour()`**
      - Set: method to set data in an object
        - » Write access: **`t1.SetHour(15)`**
  - Invalid accesses can be prevented
    - Internal information is hidden!
    - Ensures consistent state of the object!
      - Example:

```
» def SetMinute(self, minute):
      if 0 <= minute <= 59:
          self.minute = minute
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Program **`time2.py`** (part 1/4)

```
# time2.py: abstract data type for representation of time
#           (version 2)
# author: Rainer Doemer
# 02/19/04 RD    added access control methods
# 02/17/04 RD    initial version (similar to figure 7.1)

# class definition
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 SetTime(self, hour=0, minute=0, second=0):
        """sets the time of a time object"""
        self.SetHour(hour)
        self.SetMinute(minute)
        self.SetSecond(second)
...
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Program **`time2.py`** (part 2/4)

```
...
    def SetHour(self, hour=0):
        """sets the hour of a time object"""
        if (0 <= hour <= 23):
            self.__hour = hour
        else:
            raise ValueError, "Hour value out of range 0-23"

    def SetMinute(self, minute=0):
        """sets the minute of a time object"""
        if (0 <= minute <= 59):
            self.__minute = minute
        else:
            raise ValueError, "Minute value out of range 0-59"

    def SetSecond(self, second=0):
        """sets the second of a time object"""
        if (0 <= second <= 59):
            self.__second = second
        else:
            raise ValueError, "Second value out of range 0-59“
...
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Program **`time2.py`** (part 3/4)

```
...
    def GetTime(self):
        """returns the time in a tuple of (h,m,s)"""
        return (self.__hour,self.__minute,self.__second)

    def GetHour(self):
        """returns the hour of the time object"""
        return self.__hour

    def GetMinute(self):
        """returns the minute of the time object"""
        return self.__minute

    def GetSecond(self):
        """returns the second of the time object"""
        return self.__second

    def GetAMPM(self):
        """returns 'am' or 'pm' in a string"""
        if self.__hour < 12:
            return "am"
        else:
            return "pm“
...
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Program **`time2.py`** (part 4/4)

```
...
    def Print(self):
        """prints the value of a time object"""
        print "%02d:%02d:%02d" %  \
                (self.__hour,self.__minute,self.__second)

    def PrintAMPM(self):
        """prints the time in am/pm notation"""
        h = self.__hour % 12
        if h == 0:
            h = 12
        print "%2d:%02d:%02d %s" %\
                (h,self.__minute,self.__second,self.GetAMPM())
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Notes for program **`time2.py`** (1):
    - Constructor **`__init__`** takes arguments for initialization
      - Initial time can be specified at object creation
      - Default arguments are provided for convenience
    - Method **`SetTime`**
      - allows to (re-) set the time of an existing object
      - calls individual methods for data members
    - Methods **`SetHour`**, **`SetMinute`** and **`SetSecond`**
      - if argument is valid, set the internal (!) data member
      - if argument is invalid, raise an exception
    - Internal data members **`__hour`**, **`__minute`**, **`__second`**
      - marked as *private* by 2 leading underscores (__)
      - should not be accessed from outside the class
      - are subject to *name mangling* (will be renamed)

Access Control to Object Attributes

- Example: **`class Time`**
  - Notes for program **`time2.py`** (2):
    - Raising exceptions
      - Keyword **`raise`** raises the specified exception with an argument explaining the problem
      - Unless handled by an exception handler, an exception will terminate the program execution
      - More details on exceptions follow later!
    - Method **`GetTime`**
      - returns the time values in a 3-tuple (h/m/s)
    - Methods **`GetHour`**, **`GetMinute`** and **`GetSecond`**
      - return the requested time value
    - Method **`GetAMPM`**
      - returns an AM/PM indicator
    - Methods **`Print`** and **`PrintAMPM`**
      - as before, but adjusted to use modified methods

Access Control to Object Attributes

- Example: **`class Time`**
  - Interactive use of module **`time2.py`** (part 1/2)

```
% python
>>> from time2 import Time
>>> t1 = Time()
>>> t2 = Time(9,50)
>>> t1.Print()
00:00:00
>>> t2.Print()
09:50:00
>>> t1.SetTime(15,35)
>>> t1.Print()
15:35:00
>>> t1.SetSecond(45)
>>> t1.Print()
15:35:45
>>> t1.SetHour(27)
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
  File "time2.py", line 30, in SetHour
    raise ValueError, "Hour value out of range 0-23"
ValueError: Hour value out of range 0-23
...
```

Access Control to Object Attributes

- Example: **`class Time`**
  - Interactive use of module **`time2.py`** (part 2/2)

```
...
>>> t1.GetTime()
(15, 35, 45)
>>> t1.GetMinute()
35
>>> t1.GetAMPM()
'pm'
>>> t1.PrintAMPM()
 3:35:45 pm
```