CSC108: Introduction to Computer Programming

Lecture 6

Wael Aboulsaadat

Acknowledgment: these slides are based on material by: Velian Pandeliev, Diane Horton, Michael Samozi, Jennifer Campbell, and Paul Gries from CS UoT

What have we learnt up till now?

- Variables
- Logical & Mathematical Operators
- Assignment Statement
- Types & Type conversion
- if/else Statement
- print
- input & raw_input
- Functions
- Docstrings
- while loops
- Variable scope & Namespaces
- Classes & Objects

Classes & Objects (revisited)

Class

- A specification of a complex type that has
  1) attributes 2) methods

```
class Car:

    def moveBackword(self,newValue):
        self.distance = self._distance - newValue

    def getTravelledDistance(self):
        return self.distance

    def openDoor(self,nDoorNumber):
        if nDoorNumber == 1:
            self._Door1open = True
        else:
            if nDoorNumber == 2:
                self.Door2open = True
```

Class: constructor

- A constructor is a special method that has the name __init__ and is called automatically by the python interpreter on your behalf when you create an object of that class type
- This is where you put all initialization code of the attributes (aka variables) inside the object of that class type

```
class Car:
    def __init__(self):
        self._distance = 0
        self._Door1open= True
        self._Door2open= True
```

Class: self

- self refers to the address of a specific object
- It has value after you create an object of a specific type (aka class)

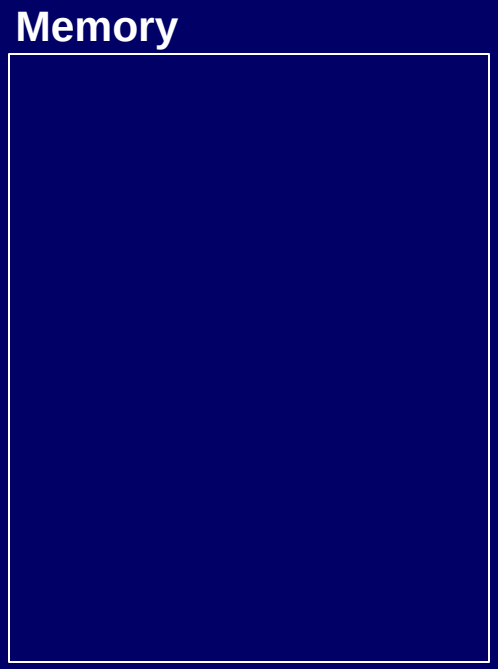
```
class Car:
    def __init__(self):
        self._distance = Num
        self._Door1open= True
        self._Door2open= True


    def moveBackword(self,newValue):
        self.distance = self.distance - newValue

    def getTravelledDistance(self):
        return self._distance
    .....
```

Object

- An instantiation of a class. Each instance is called an object.
- We can create many instances (aka objects) of the same class (aka type)

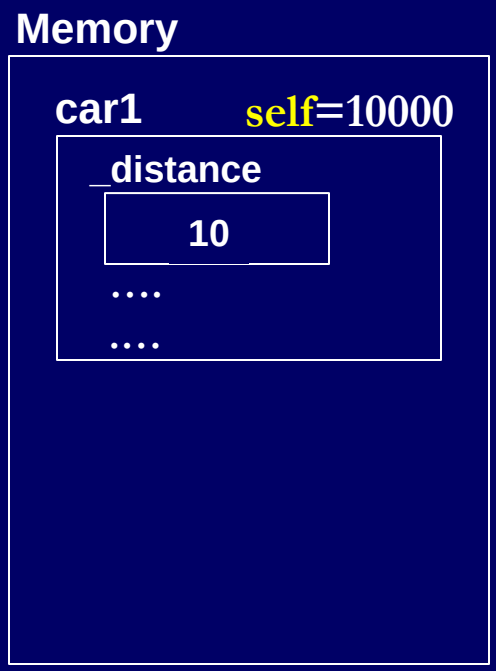


Object

- An instantiation of a class. Each instance is called an object.

- We can create many instances (aka objects) of the same class (aka type)

```
car1 = Car( )
car1.moveForward(10)
print car1.getTravelledDistance( )
```



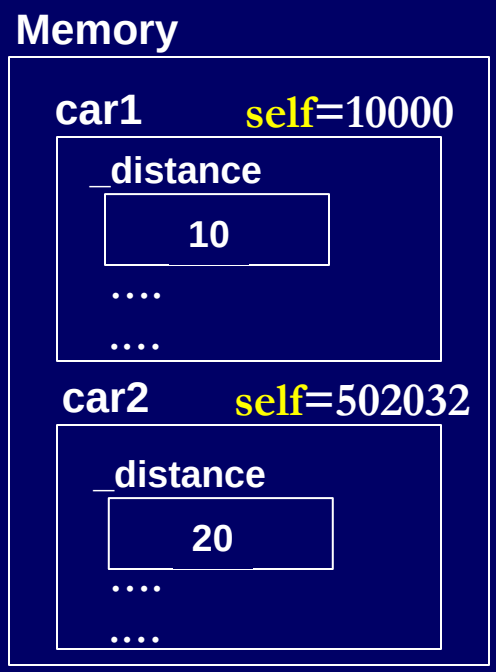
Object

- An instantiation of a class. Each instance is called an object.

- We can create many instances (aka objects) of the same class (aka type)

```
car1 = Car( )
car1.moveForward(10)
print car1.getTravelledDistance( )

car2 = Car( )
car2.openDoor( 1 )
ca2.moveForward( 20 )
print car2.getTravelledDistance( )
```



Mutability

- Some objects in Python are immutable, meaning that once created, their values cannot be changed.
- Objects of types int, bool, double, float and str are immutable.
- Some objects are mutable, meaning that their values can be changed in-place without creating new objects.
- Objects of type list are mutable.

Immutable Objects

- Any method that 'changes' the values of immutable objects is discarding them and creating new ones.

```
>>> x = ‘Hello World'
>>> id(x)
4578104
>>> x = x.upper()
>>> x
‘HELLO WORLD'
>>> id(x)
4578024
```

Mutable Objects

- In contrast, mutable objects are changed in place:

```
>>> x = [1,2,3]
>>> id(x)
4576400
>>> x.append(4)
>>> x
[1, 2, 3, 4]
>>> id(x)
4576400
```

Mutable Objects

- In contrast, mutable objects are changed in place:

```
>>> x = [1,2,3]
>>> id(x)
4576400
>>> x.append(4)
>>> x
[1, 2, 3, 4]
>>> id(x)
4576400
```

Mutability and Slicing

- Lists & Strings can be sliced
- listname[a:b] - *returns a list composed of* items from index a to b-1
- listname[a:] - *returns a list composed of items* from index a to the end of the list
- listname[:a] - *returns a list composed of items* from index 0 to a-1
- listname[:] - *returns a copy of the list*

Mutability and Slicing

- Slicing all return a copy of the object. It causes the creation of a copy and modifying the copy.

```
>>> lst=[1,2,2,3,4,54,5]
>>> x = lst[1:5]
>>> print x
[2, 2, 3, 4]
>>> id(lst)
13069928
>>> id(x)
13069968
```

Mutability and Aliasing

- Since giving values to parameters is a form of assignment, if you have a mutable parameter, it will refer to the same object as the argument, and changing one will change the other.

```
def add_4_5(alist):
        print alist.extend([4,5])
```

```
>>> x = [1,2,3]
>>> add_4_5(x)
>>> print x
[1,2,3,4,5]
```

Files

Reading and Writing Files

- Creating and opening a file object

fileObject = open(filePath, mode)

fileObject – Contains the pointer to file object created
filePath – Location of file in OS file system (relative or absolute)
mode – Read (‘r’), Write(‘w’), Append (‘a’), R/W (‘r+’)
mode (binary) – ‘rb’, ‘wb’, ‘r+b’, ‘ab’

Difference between text and binary files

- At heat, all files are binary files!
- Text files uses ASCII codes in storing information
- Binary files do no use ASCII codes in storing information

Text file example

Hi there!,

This is an example of a text file. Emails, many internet documents are some of the examples of information stored as text.

Cheers!
Wael

What you see on screen

72 105 32 116 104 101 114 101 33 44 10 10 84 104 105 115
32 105 115 32 97 110 32 101 120 97 109 112 108 101 32 111
102 32 32 97 32 116 101 120 116 32 102 105 108 101 46 32
69 109 97 105 108 115 44 32 109 97 110 121 32 105 110
116 101 114 110 101 116 32 100 111 99 117 109 101 110 116
115 32 97 114 101 32 115 111 109 101 32 111 102 32 116
104 101 32 101 120 97 109 112 108 101 115 32 111 102 32
105 110 102 111 114 109 97 116 105 111 110 32 115 116 111
114 101 100 32 97 115 32 116 101 120 116 46 10 10 67 104
101 101 114 115 33 10 87 97 101 108 10 10

ASCII equivalent

10101010101010101010101111101010101010111100000110101010101010101010101010101010…….

What's stored on disk

Try
http://www.cs.carleton.edu/faculty/adalal/teaching/f05/107/applets/ascii.html

Text file example

Hi there!,

This is an example of a text file. Emails, many internet documents are some of the examples of information stored as text.

Cheers!
Wael

What you see on screen

72 105 32 116 104 101 114 101 33 44 10 10 84 104 105 115
32 105 115 32 97 110 32 101 120 97 109 112 108 101 32 111
102 32 32 97 32 116 101 120 116 32 102 105 108 101 46 32
69 109 97 105 108 115 44 32 109 97 110 121 32 105 110
116 101 114 110 101 116 32 100 111 99 117 109 101 110 116
115 32 97 114 101 32 115 111 109 101 32 111 102 32 116
104 101 32 101 120 97 109 112 108 101 115 32 111 102 32
105 110 102 111 114 109 97 116 105 111 110 32 115 116 111
114 101 100 32 97 115 32 116 101 120 116 46 10 10 67 104
101 101 114 115 33 10 87 97 101 108 10 10

ASCII equivalent

10101010101010101010101111101010101010111100000110101010101010101010101010101010…….

What's stored on disk

Try
http://www.cs.carleton.edu/faculty/adalal/teaching/f05/107/applets/ascii.html

Text file example

Hi there!,

This is an example of a text file. Emails, many internet documents are some of the examples of information stored as text.

Cheers!
Wael

What you see on screen

72 105 32 116 104 101 114 101 33 44 10 10 84 104 105 115
32 105 115 32 97 110 32 101 120 97 109 112 108 101 32 111
102 32 32 97 32 116 101 120 116 32 102 105 108 101 46 32
69 109 97 105 108 115 44 32 109 97 110 121 32 105 110
116 101 114 110 101 116 32 100 111 99 117 109 101 110 116
115 32 97 114 101 32 115 111 109 101 32 111 102 32 116
104 101 32 101 120 97 109 112 108 101 115 32 111 102 32
105 110 102 111 114 109 97 116 105 111 110 32 115 116 111
114 101 100 32 97 115 32 116 101 120 116 46 10 10 67 104
101 101 114 115 33 10 87 97 101 108 10 10

ASCII equivalent

10101010101010101010111111010101010111110000011010101010101010101010101010…….

What's stored on disk

Try
http://www.cs.carleton.edu/faculty/adalal/teaching/f05/107/applets/ascii.html

Binary file example

10101010101010101010110101111111010101010

What you see on screen

What's stored on disk

Try
http://www.cs.carleton.edu/faculty/adalal/teaching/f05/107/applets/ascii.html

Reading Files

- The “fileObject.read(size)” method reads the indicated number of bytes into an string
  - If size < 0 or omitted, the whole file is read

```
>>> fileObj = open('IO/ConfigFile.txt', 'r')

>>> print fileObj
<open file 'IO/ConfigFile.txt', mode 'r' at 0x00A8F8D8>

>>> print fileObj.read()
posx = 10
posy = 20
posz = 30

>>> fileObj.close()
```

Reading Files

- The readline method reads one line at a time
- The readlines method reads all lines in a file on a string list
- A for loop can be used to iterate line by line in a file

```
for line in fileObj:
    print line
```

Writing Files

- The write method writes a given string into a file
  - fileObj.write(string)
  - Everything written to a file must be or casted to string
  - Values read from files, if not strings, should be converted to their corresponding type

3 Steps while Working with Files

- open
- Process (read or write)
- close

Files example

What have we learnt today?

- Mutability
- Files

This Week's To Do List

- Go through lecture slides – make sure you try the code snippets
- Try the lecture's programs posted on course website