

Inheritance

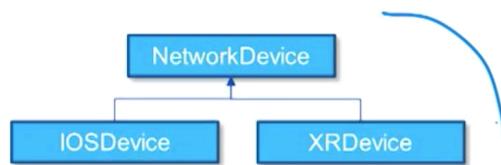
- Parent or base class

Contains basic attributes common to all objects that *inherit* from that base class.

- Child or sub class

Contains values that are specific to that type of object

- Relationship: "is-a"



```
class NetworkDevice():
    # general data: ip,user,pw
    # general methods

class IOSDevice(NetworkDevice):
    # IOS-specific data
    # IOS-specific methods

class XRDevice(NetworkDevice):
    # XR-specific data
    # XR-specific methods

...
```



© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 67

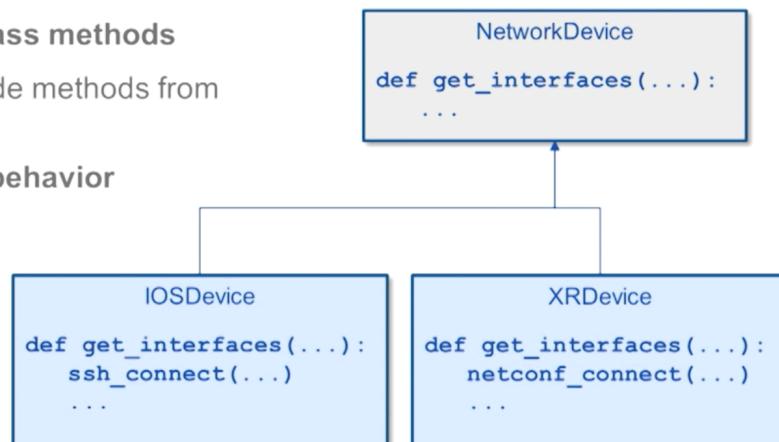
Overriding Methods

- Overriding base-class methods

Subclasses can override methods from base class if required

- Subclass-specific behavior

Allows subclasses to implement specific behavior unique to that subclass



© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 68

1.5x

Composition

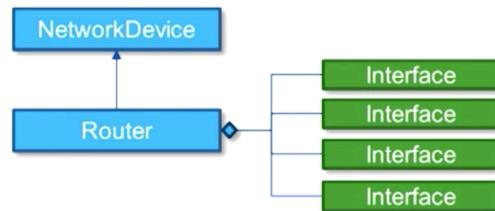
- Alternative to *inheritance*

Don't overuse inheritance, where *composition* would be simpler.

- **Classes can contain other classes**

Classes define attributes which are classes themselves.

- Relationship: '**has-a**'



Examples:

- A router *is a* network device
- A router *has a* set of interfaces
- A wireless controller *is a* controller
- A wireless controller *has a* set of wireless APs that it controls



© 2016 Cisco and/or its affiliates. All rights reserved. Cisco Confidential 69

-7:03 1.5x ⏪ ⏹ ⏷

One of the more significant motivations for the move to object-oriented programming is inheritance. OOP attempts to model the 'real world', and in the real world there are types of things that inherit attributes from other types of things. For example, a circle is a type of shape, as is a rectangle or an octagon or oval. All these shapes have certain attributes in common, such as the ability to calculate the area of the shape..

In the networking world, there are network nodes. Different types of nodes would be hosts and network devices. Different types of network devices would be routers, switches, wireless access points, and so on.

This idea of certain types having 'child' classes, which inherit the attributes of its 'parent class, is fundamental to object-oriented programming. With Python, you can define classes, and sub-classes which inherit properties of the base or parent class.

This lesson will focus on how the OOP principle of inheritance is implemented using Python.

```
cisco@cisco-python:/var/local/PyNE/labs/sections/section13$ cat devices
d01-ios,ios,10.3.21.5,cisco,cisco
d02-ios,ios,10.3.21.6,cisco,cisco
d03-nx(nx-os,10.3.21.7,cisco,cisco
d04-nx(nx-os,10.3.21.8,cisco,cisco
d05-xr,ios-xr,10.3.21.9,cisco,cisco
d06-xr,ios-xr,10.3.21.10,cisco,cisco
d07-xe,ios-xe,10.3.21.19,cisco,cisco
d08-xe,ios-xe,10.3.21.22,cisco,cisco
```



```

        else:
            continue # go to the next device in the file

        devices_list.append(device) # add this device object to list

    file.close() # Close the file since we are done with it

    return devices_list

----- Function to go through devices printing them to table -----
def print_device_info(devices_list):

    print ''
    print 'Name      OS-type  IP address      Username  Password'
    print '-----  -----  -----  -----  -----'

    # Go through the list of devices, printing out values in nice format
    for device in devices_list:

        print '{0:8} {1:8} {2:16} {3:8} {4:8}'.format(device.name,
                                                       device.os_type,
                                                       device.ip_address,
                                                       device.username,
                                                       device.password)

    print ''

----- Main: read device info, then print -----
devices = read_device_info('devices')
print_device_info(devices)

```

This will only print the IOS and XR devices

```

cisco@cisco-python:/var/local/PyNE/labs/sections/section13$ python S07-3-inheritance.py
Name      OS-type  IP address      Username  Password
-----  -----  -----  -----  -----
d01-ios   ios      10.3.21.5     cisco     cisco
d02-ios   ios      10.3.21.6     cisco     cisco
d05-xr    ios-xr   10.3.21.9     cisco     cisco
d06-xr    ios-xr   10.3.21.10    cisco     cisco

```