

LAB – CREATING CLASSES AND OBJECTS**OBJECTIVE**

In this lab, you will define classes and objects.

PART 1

Open a terminal and switch to the lab directory

STEP 1: OPEN A TERMINAL WINDOW

Double-click the Terminal icon on the desktop to open the terminal window for use in this lab.

STEP 2: CHANGE DIRECTORY

Change to the directory **labs/prne/** in the user home directory, which holds the files for the course labs.

```
~$ cd labs/prne/
```

PART 2

Open **Visual Studio Code**, create a new file and save it with a filename of **creating-classes-and-objects-part-2.py**, ensuring to save the file in the **~/labs/prne/** directory.

This python application will:

- Define a class
- Use that class definition to create objects to store information about networking devices
- Display the information for those network device objects

NOTE:

For this lab, you will hard-code your device information as you create your network device objects

STEP 1: DEFINE CLASS

Define a class called **NetworkDevice** and define a method within the class that takes device information as parameters: device name, OS-type, IP address, username, and password. Allow the username and password to be omitted, providing default values of **prne** and **cisco**.

```
# Class to hold information about a network device
class NetworkDevice():

    def set_info(self, name, os, ip, user='prne', pw='cisco'):
        self.name = name
        self.ip_address = ip
        self.os_type = os
        self.username = user
        self.password = pw
```

STEP 2: CREATE FUNCTION TO DISPLAY INFORMATION

Define a function to display a table of device information (name, OS-type, IP, username, password) for every device. Pass in a list of devices, where each device is an object of type **NetworkDevice**.

```
# 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('{0:11} {1:8} {2:17} {3:9} {4:9}'.format(device.name,
                                                       device.os_type,
                                                       device.ip_address,
                                                       device.username,
                                                       device.password))

    print('')
```

STEP 3: CREATE MAIN CODE

Your **main** code should create two or more **NetworkDevice** objects. For each object, call your method to set the device information.

Note: Since you are hard coding these devices, you are not reading from a file, or using a loop. Create the first object and set its info, then create the second object and set its info.

After creating your device objects, add them to a Python list of devices. Call your print function, passing in the devices list.

```
# ---- Main: read device info, then print -----
dev1 = NetworkDevice()
dev1.set_info('dev1', 'IOS-NX', '203.0.113.253')

dev2 = NetworkDevice()
dev2.set_info('dev2', 'IOS-XE', '198.51.100.253', 'chuck', 'secret')

print_device_info([dev1, dev2])
```

STEP 4: SAVE, RUN AND VERIFY APPLICATION

Save your application and then run it from the terminal rather than from within visual studio code.

```
~/labs/prne$ python3 creating-functions-that-return-values-part-2.py
```

The output from your application will be displayed in your terminal window, verify that it is comparable to below.

```
devasc@labvm:~/labs/prne$ python3 creating-classes-and-objects-part-2.py
```

Name	OS-type	IP address	Username	Password
-----	-----	-----	-----	-----
dev1	IOS-NX	203.0.113.253	prne	cisco
dev2	IOS-XE	198.51.100.253	chuck	secret

PART 3

Open **Visual Studio Code**, create a new file and save it with a filename of **creating-functions-that-return-values-part-3.py**, ensuring to save the file in the **~/labs/prne/** directory.

This python application will define a network device class, with an initialization method for setting attributes for each created object. Device information will be read from the files **devices-10.txt** and **devices-11.txt**.

STEP 1: DEFINE CLASS

Define a class called **NetworkDevice**. Define an initialization method (called **__init__**) within the class that takes device information as parameters (device name, OS-type, IP address, username, and password. Remember that 'self' must be the first parameter for every method.

```
# Class to hold information about a network device
class NetworkDevice():

    def __init__(self, name, ip, os, user='prne', pw='cisco'):
        self.name = name
        self.ip_address = ip
        self.os_type = os
        self.username = user
        self.password = pw
```

STEP 2: CREATE FUNCTION TO READ INFORMATION FROM FILE

Create a function that takes the name of the devices file as input, reads the device information from the file, and creates network device objects, adding them to a list of devices. The result will be a list of network device objects, based on the information read from the file. The function should return the list of devices to the caller.

```
# Function to read device information from file
def read_device_info(devices_file):

    # Create a list for all devices
    devices = []

    # Read in the devices from the file
    file = open(devices_file, 'r')
    for line in file:

        # Get device info into list
        device_info = line.strip().split(',')

        # Create a device object with this data
        device = NetworkDevice(device_info[0], device_info[2],
                               device_info[1], device_info[3],
                               device_info[4])

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

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

    # return a reference to the list we created
    return devices
```

STEP 3: CREATE FUNCTION TO DISPLAY INFORMATION

Create a print function that takes as input a list of network device objects, and displays a table of the devices from the 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, displaying values in nice format
    for device in devices_list:

        print('{0:11} {1:8} {2:16} {3:9} {4:9}'.format(device.name,
                                                        device.os_type,
                                                        device.ip_address,
                                                        device.username,
                                                        device.password))

    print('')
```

STEP 4: CREATE MAIN CODE

Create main application code that calls the function that reads device information from the **devices-10.txt** file and then displays the information, and then repeat for the **devices-11.txt** file.

```
# Main: read device info, then print
devices_list = read_device_info('devices-10.txt')
print_device_info(devices_list)

devices_list = read_device_info('devices-11.txt')
print_device_info(devices_list)
```

STEP 5: SAVE, RUN AND VERIFY APPLICATION

Save your application and then run it from the terminal rather than from within visual studio code.

```
~/labs/prne$ python3 creating-functions-that-return-values-part-3.py
```

The output from your application will be displayed in your terminal window, verify that it is comparable to below.

```
devasc@labvm:~/labs/prne$ python3 creating-classes-and-objects-part-3.py
```

Name	OS-type	IP address	Username	Password
d01-is	ios	192.168.122.1	cisco	cisco
d02-is	ios	192.168.122.2	cisco	cisco
d03-nx	nx-os	192.168.122.3	cisco	cisco
d04-nx	nx-os	192.168.122.4	cisco	cisco
d05-xr	ios-xr	192.168.122.5	cisco	cisco
d06-xr	ios-xr	192.168.122.6	cisco	cisco
d07-xe	ios-xe	192.168.122.7	cisco	cisco
d08-xe	ios-xe	192.168.122.8	cisco	cisco

Name	OS-type	IP address	Username	Password
Rtr1-Prne	ios	10.50.100.1	prne	cisco
Rtr2-Prne	ios	203.0.113.1	prne	cisco
Rtr3-Prne	ios	198.51.100.1	prne	cisco

PART 4 (OPTIONAL BUT HIGHLY RECOMMENDED)

As this lab is completed in NETLAB+ and your code files will be erased when the reservation ends, it is advisable to save your files in GitHub under your repository for this course.