LAB – USING BREAKS AND CONTINUE

OBJECTIVE

In this lab, you will use the break and continue statements to modify loop behaviour.

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 **using-breaks-and-continue-part-2.py**. Ensuring to save the file in the **~/labs/prne/** directory, as otherwise the code will require modification to find the associated files that are used.

This python application will read the information from the **devices-07.txt** file and highlight any duplicate IP addresses found. In order to write this application, you will:

- Use a for loop to read in device information
- Use a set to store known IP addresses
- Do a comparison of each device IP address against a set of known IP addresses.
- Use a continue statement to move to the next iteration if a match is found.

STEP 1: CREATE EMPTY LIST

Create an empty list called device_list.

```
# Create the outer list for all devices
devices_list = []
```

STEP 2: READ DEVICES INFORMATION

Read device information from the file and store in the list devices_list.

```
# Read in the devices from the file
file = open('devices-07.txt', 'r')
for line in file:

    device_info_list = line.split(',')  # Get device info into list
    devices_list.append(device_info_list)
```

STEP 3: DISPLAY TABLE HEADING

Display a heading for the table of device information that will be read, to include device name, OS type, IP address and software version.

```
# Display heading
print('')
print('Name     OS-type    IP address           Software            ')
print('-------  ---------  -------------------  --------------------')
```

STEP 4: CREATE EMPTY SET

Create an empty collection that holds the IP addresses with no duplicates.

```
# Create unordered collection with no duplicate elements
ip_addresses = set()
```

STEP 5: DISPLAY DEVICE INFORMATION IN TABLE

Display the device information in a table and if an IP address is a duplicate of one seen before, print a message indicating it is a duplicate.

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

    print('{0:8} {1:10} {2:20} {3:20}'.format(device[0], device[1],
                                              device[2], device[3]),
                                              end=' ')

    # Display 'duplicate' if IP address exists for another device
    if device[2] in ip_addresses:
        print('*Duplicate IP!*')
        continue

    ip_addresses.add(device[2])
    # Return to a new line
    print('')

# Display a blank line to make easier to read
print('')
```

STEP 6: CLOSE FILE

Close the file.

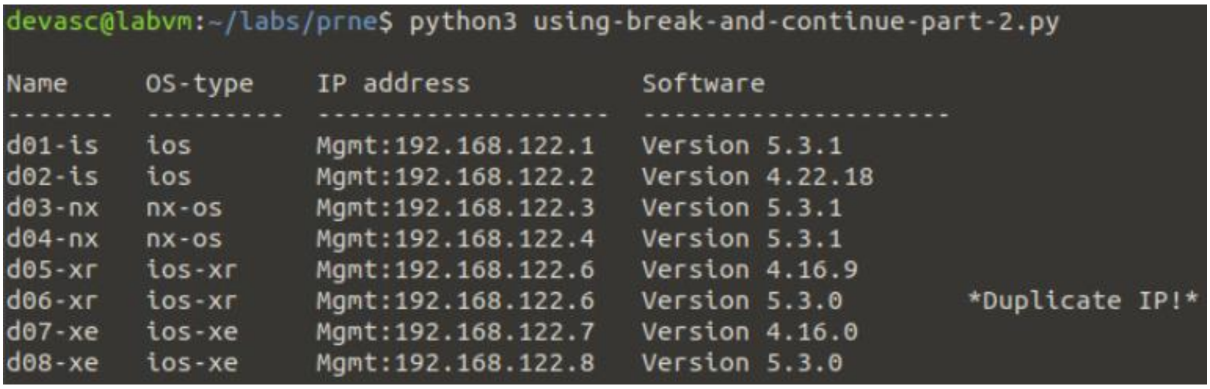
```
# Close the file
file.close()
```

STEP 7: SAVE, RUN AND VERIFY APPLICATION

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

```
~/labs/prne$ python3 using-breaks-and-continue-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 using-break-and-continue-part-2.py

Name     OS-type    IP address           Software
-------  ---------  -------------------  --------------------
d01-is   ios        Mgmt:192.168.122.1   Version 5.3.1
d02-is   ios        Mgmt:192.168.122.2   Version 4.22.18
d03-nx   nx-os      Mgmt:192.168.122.3   Version 5.3.1
d04-nx   nx-os      Mgmt:192.168.122.4   Version 5.3.1
d05-xr   ios-xr     Mgmt:192.168.122.6   Version 4.16.9
d06-xr   ios-xr     Mgmt:192.168.122.6   Version 5.3.0         *Duplicate IP!*
d07-xe   ios-xe     Mgmt:192.168.122.7   Version 4.16.0
d08-xe   ios-xe     Mgmt:192.168.122.8   Version 5.3.0
```

PART 3

Open **Visual Studio Code**, create a new file and save it with a filename of **using-breaks-and-continue-part-3.py**. Ensuring to save the file in the **~/labs/prne/** directory, as otherwise the code will require modification to find the associated files that are used. This python application will allow a user to search for device information by entering the IP address. If the IP address is found, the application will display the device information. The application will iterate until the user enters Ctrl-C.

In order to write this application, you will:

- Use a for loop to read in device information
- Use a while loop to allow users to input IP addresses to search for
- Use range to iterate through devices in the list
- Use continue to continue searching the list
- Use break to exit when user presses Ctrl-C
- Use for...else to print if the list was exhausted without finding a match

STEP 1: CREATE EMPTY LIST

Create an empty list called device_list.

```
# Create the outer list for all devices
devices_list = []
```

STEP 2: DISPLAY TABLE HEADING

Display a heading for the table of device information that will be read, to include device name, OS type, IP address and software version.

```
# Display heading
print('')
print('Name     OS-type    IP address           Software            ')
print('-------  ---------  -------------------  --------------------')
```

STEP 3: CREATE INDEX NUMBER

Create the start number for the index.

```
# Start index number
index = 0
```

STEP 4: READ DEVICES INFORMATION

Read device information from the file and store in the list devices_list, whilst displaying a table in the terminal.

```
# Read in the devices from the file
file = open('devices-07.txt', 'r')
for line in file:

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

    print('{0:2}: {1:8} {2:8} {3:20} {4:20}'.format(index+1,
                                                    device_info[0],
                                                    device_info[1],
                                                    device_info[2],
                                                    device_info[3]))

    index += 1  # increment our index

# Display a blank line to make easier to read
print('')
```

STEP 5: CREATE LOOP

Create a loop that continues forever (while true) reading user input, and matching their input IP address with an item from the list of devices. If the user enters Ctrl-C, exit the program. If the user enters anything else, use it to search for a matching IP address from the list of devices. Using a for statement with a range. The range will go from 0 to the length of the list. If the IP address is found, print a message with the device data. If the IP address is not found, print a message stating: "--- Given IP address not found ---".

```
while True:  # Loop forever, until user terminates program

    # Request user to input the IP address we will search for
    try:
        ip_address = input('Enter device IP address to find (Ctrl-C to 
exit): ')
    except KeyboardInterrupt:
        break

    # Loop through our devices looking for a match on IP address
    for index in range(0, len(devices_list)):

        # Get information for this device in the list
        device_info = devices_list[index]

        # Check to see if device IP is a match
        if device_info[2][5:] == ip_address:

            # If a match, print results and stop looking
            print('{0:2}: {1:8} {2:8} {3:20} {4:20}'.format(index+1, 

device_info[0], 

device_info[1], 

device_info[2], 

device_info[3]))
            # Display a blank line to make easier to read
            print('')
            break
        else:
            continue

    else:  # If we exhausted the device list, IP not found
        print('--- Given IP address not found ---')
        # Display a blank line to make easier to read
        print('')

# Display a blank line to make easier to read
print('')
print('Device search terminated.\n')
```

STEP 6: CLOSE FILE

Close the file.

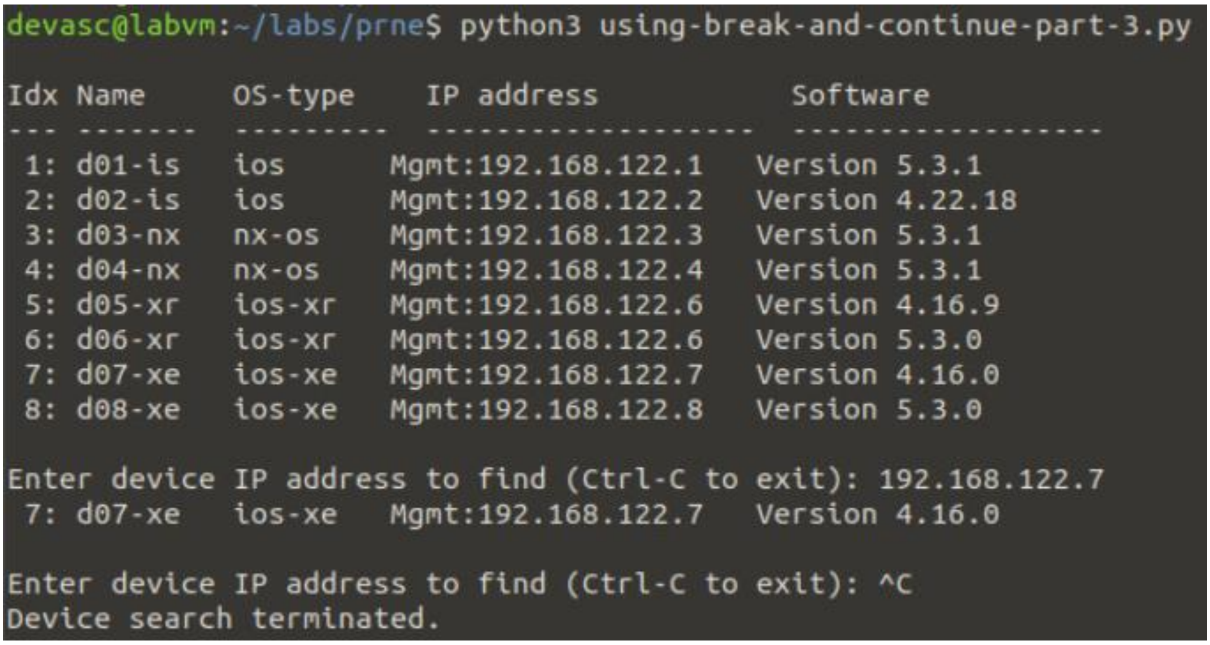
```
# Close the file
file.close()
```

STEP 5: SAVE, RUN AND VERIFY APPLICATION

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

```
~/labs/prne$ python3 using-breaks-and-continue-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 using-break-and-continue-part-3.py

Idx Name     OS-type    IP address          Software
--- -------  ---------  ------------------  ------------------
  1: d01-is   ios      Mgmt:192.168.122.1   Version 5.3.1
  2: d02-is   ios      Mgmt:192.168.122.2   Version 4.22.18
  3: d03-nx   nx-os    Mgmt:192.168.122.3   Version 5.3.1
  4: d04-nx   nx-os    Mgmt:192.168.122.4   Version 5.3.1
  5: d05-xr   ios-xr   Mgmt:192.168.122.6   Version 4.16.9
  6: d06-xr   ios-xr   Mgmt:192.168.122.6   Version 5.3.0
  7: d07-xe   ios-xe   Mgmt:192.168.122.7   Version 4.16.0
  8: d08-xe   ios-xe   Mgmt:192.168.122.8   Version 5.3.0

Enter device IP address to find (Ctrl-C to exit): 192.168.122.7
  7: d07-xe   ios-xe   Mgmt:192.168.122.7   Version 4.16.0

Enter device IP address to find (Ctrl-C to exit): ^C
Device search terminated.
```

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.