***It is quick (four questions).  
  
Also if you valued this course and want to provide us with a testimonial--email us over your thoughts. Our team is small so testimonials help.  
  
Thanks!***  
***-------------***  
  
  
***Note: There is a table of contents for each video located at the bottom of this email including timestamps. This should be helpful in navigating the videos.***  
  
  
﻿In this email of Learning Python we are going to cover the following:

1. **Classes and Objects Overview**  
   Video link [https://vimeo.com/845800025](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDU4MDAwMjU_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.960HWEAsUser4LKNNsvaUHhZc_P_D_d0qHubtOdMetA)  
   Length is 4 minutes
2. **Dunder-init and Creating Objects**  
   Video link [https://vimeo.com/845991911](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDU5OTE5MTE_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.21k9_hTlxvFuJC-MB_gL4Lj23W96aoOjZozR0-qvino)  
   Length is 11 minute
3. **Methods**  
   Video link [https://vimeo.com/845993458](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDU5OTM0NTg_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.c98h6q-W3jkolaQO3R7nzlrkp5vz9VFIstrHRtKQ6Fc)  
   Length is 7 minutes
4. **Static Methods**(\*optional/intermediate content)  
   Video link [https://vimeo.com/848507954](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDg1MDc5NTQ_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.iTNXWa48TfDb0lm2budPhR2H3ZU-EhS0gVEHaN5eDUU)  
   Length is 11 minutes
5. **Special Names**  
   Video link [https://vimeo.com/848859608](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDg4NTk2MDg_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.PucQfRcLDA7oV1F-THMY1cgcAklUV_QMtUTJQLfCKLc)  
   Length is 6 minutes
6. **Properties**(\*optional/intermediate content)  
   Video link [https://vimeo.com/849313005](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDkzMTMwMDU_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.7Eu8IVbpmW-dz4ppzo0VBE7UOo_kzwa9gsSKEuU7n1Q)  
   Length is 7 minutes
7. **Properties - Setters**(\*optional/intermediate content)  
   Video link [https://vimeo.com/849318701](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDkzMTg3MDE_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.HbDGZDcf0k-dg-jMXyB2_F4mTWFxg2boKSCN3zzk7NU)  
   Length is 2 minutes
8. **Property Example**(\*optional/intermediate content)  
   Video link [https://vimeo.com/849320260](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDkzMjAyNjA_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.2NZw9pY8Pc1DyvEkSaSBOVKak9XV5DrfPerTce8rkDQ)  
   Length is 3 minutes
9. **Deleters**(\*optional/intermediate content)  
   Video link [https://vimeo.com/849538605](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3ZpbWVvLmNvbS84NDk1Mzg2MDU_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.7x1Gtk5LEcAZ4eZiWwajZudtNNoHuMGzWA_wOQa2TLo)  
   Length is 1 minutes

**\* Optional/intermediate content - this content is more intermediate and can be safely skipped.**In other words, feel free to skip this content if you are time limited or are struggling with the more fundamental content.  
  
  
  
**Collateral Material (programs used in the videos):**  
  
*In collateral, I try to provide the Python programs shown in the videos. In other words, I try to make it easy for you to reproduce the examples from the videos.*  
  
[Lesson7 Collateral Programs](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..rGSti0yrdaF8RBfqQ4QOaHAG9qsOemLb6apaRlEbrUg)  
  
  
  
  
**Additional Content:**  
  
[Object-Oriented Programming (OOP) in Python 3](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL3JlYWxweXRob24uY29tL3B5dGhvbjMtb2JqZWN0LW9yaWVudGVkLXByb2dyYW1taW5nLz9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.4hil5150QXk383lGlfMMZwFZnIExAELX2g0T6hLKQw8)  
Real Python article that discusses class and object basics. This article covers class syntax, dunder-init, instance and class attributes, creating objects, and using methods. It also covers inheritance basics including overriding parent methods and super().  
  
[Classes and Objects — the Basics](http://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9..CrvXMnNYMu65hC6ofYuisI4gGDyHl64pAhclbpRSEQE)  
From the book, "How to Think Like a Computer Scientist: Learning with Python 3", this section discusses classes and objects. It covers dunder-init, instance attributes, and methods.  
  
[The Meaning of Underscores in Python](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2RiYWRlci5vcmcvYmxvZy9tZWFuaW5nLW9mLXVuZGVyc2NvcmVzLWluLXB5dGhvbj9fX3M9bXpwbTZucTNjbXR6eDAzdGs1ZGcifQ.2M0GPk4GOH-06TxNBUoaKslJTM6ufzyP3-tUt5QRrmc)  
Article from David Bader covering some special names in Python. **Note, the very first time you go to this website, it has a "Python Tricks" email-list it asks you to join. You can bypass this by clicking the "x" in the upper right corner.**This article covers private attributes, name mangling, and dunder-methods (as well as some other special names).   
  
  
  
  
**Exercises**

Reference code for these exercises is posted on GitHub at:

[https://github.com/twin-bridges/learning\_python/tree/main/lesson7/exercises](https://t.dripemail2.com/c/eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJkZXRvdXIiLCJpc3MiOiJtb25vbGl0aCIsInN1YiI6ImRldG91cl9saW5rIiwiaWF0IjoxNzU5ODM2MzQ0LCJuYmYiOjE3NTk4MzYzNDQsImFjY291bnRfaWQiOiI0MjU0NDk3IiwiZGVsaXZlcnlfaWQiOiJjczVtOW1ucHVlYno0dThtOGhycSIsInRva2VuIjoiY3M1bTltbnB1ZWJ6NHU4bThocnEiLCJzZW5kX2F0IjoxNzU5ODQ5MjAwLCJlbWFpbF9pZCI6NzUwNDk2NywiZW1haWxhYmxlX3R5cGUiOiJDYW1wYWlnbiIsImVtYWlsYWJsZV9pZCI6NDc4NjE0LCJ1cmwiOiJodHRwczovL2dpdGh1Yi5jb20vdHdpbi1icmlkZ2VzL2xlYXJuaW5nX3B5dGhvbi90cmVlL21haW4vbGVzc29uNy9leGVyY2lzZXM_X19zPW16cG02bnEzY210engwM3RrNWRnIn0.uXzpwMQqWoSeNrED0ADy3NVMpfgnky3DAjPootJjEuo)

1. Create a NetworkDevice class that accepts four arguments: host, platform, username, and password. Assign these arguments to the object inside of dunder-init() method. In other words, inside of dunder-init(), you should have code that does the following (for all four of the attributes):

self.host = host

Create a dunder-str() method that returns a string representation of the object and includes both the host attribute and the platform attribute in said string.  
  
From this class, create two NetworkDevice objects and print the objects out to standard output.  
  
Your output should look similar to the following:

NetworkDevice: [host1.domain.com](http://host1.domain.com/) (cisco\_xe)

NetworkDevice: [host2.domain.com](http://host2.domain.com/) (juniper\_junos)

2. Create an Interface class that accepts the following five arguments: intf\_name, intf\_mode, access\_vlan, speed, and duplex. Assign these arguments to the object inside of dunder-init() method. Your dunder-init() method should have default values for the following parameters intf\_mode="access", access\_vlan=1, speed="1Gbps", duplex="full" (so only "intf\_name" does not have a default value).  
  
The definition of your dunder-init() method should look similar to the following:

def \_\_init\_\_(

self,

intf\_name,

intf\_mode="access",

access\_vlan=None,

speed="1Gbps",

duplex="full",

):

Inside of your dunder-init() method you should assign all of the attributes to the corresponding argument values.  
  
Additionally, your dunder-init() method should contain certain checks. For "intf\_mode", your dunder-init() method should check that the value is either "access" or "trunk". If the value is neither of these, then you should raise a ValueError exception.  
  
If the "intf\_mode" is access, then you should also check that the "access\_vlan" is an integer. If the intf\_mode is trunk, then you should set the "access\_vlan" attribute to None.  
  
Additionally, create a dunder-str() method that will display details about the interface when the object is printed.  
  
Initialize seven Interface objects with the details specified below (the output of your program should also be similar to this):

Interface: Et1 (1Gbps/full, Mode: access, Vlan: 1)

Interface: Et2 (1Gbps/full, Mode: access, Vlan: 2)

Interface: Et3 (1Gbps/full, Mode: access, Vlan: 3)

Interface: Et4 (1Gbps/full, Mode: access, Vlan: 4)

Interface: Et5 (1Gbps/full, Mode: access, Vlan: 5)

Interface: Et6 (1Gbps/full, Mode: access, Vlan: 6)

Interface: Et7 (1Gbps/full, Mode: trunk)

3a. Create an "OSPFRouter" class that represents one router running OSPF. This OSPFRouter class should accept the following arguments in dunder-init(): instance\_id, area\_id, router\_id, is\_dr, and is\_bdr. All of these arguments should be bound as attributes to the object. Both the "is\_dr" and the "is\_bdr" parameters should default to False (in the dunder-init() parameter definition).  
  
Additionally, create a private attribute named \_neighbors in dunder-init(). This attribute should be initialized to an empty set.  
  
3b. Add a dunder-str() method that prints out a representation of a OSPFRouter object. This string representation should include all of the above attributes (including \_neighbors).  
  
3c. Add an add\_neighbor() method and a remove\_neighbor() method that adds and removes neighbors from the \_neighbors set.  
  
3d. Create an instance of this OSPFRouter class representing the following OSPF state:

arista2#show ip ospf database

OSPF Router with ID(10.220.88.29) (Instance ID 42) (VRF default)

Router Link States (Area 0.0.0.0)

Link ID ADV Router Age Seq# Checksum Link count

10.220.88.28 10.220.88.28 582 0x80000008 0xa410 1

10.220.88.30 10.220.88.30 307 0x80000006 0xa40c 1

10.220.88.32 10.220.88.32 297 0x80000008 0x9c0c 1

10.220.88.34 10.220.88.34 292 0x80000006 0x9c08 1

10.220.88.31 10.220.88.31 305 0x80000005 0xa40a 1

10.220.88.33 10.220.88.33 292 0x80000006 0x9e09 1

10.220.88.29 10.220.88.29 581 0x80000007 0xa40e 1

10.220.88.35 10.220.88.35 287 0x80000006 0x9a07 1

**Note, Arista2 is the designated router for this area (Area 0).**  
  
  
4. Update your NetworkDevice class from exercise1 by converting the password attribute to a private attribute (named \_password).  
  
Add an @property for the 'password' attribute such that you retrieve the private password attribute and convert all of its characters over to "\*" characters. Consequently, whenever anyone accesses nd\_obj.password all they will see is "\*" characters (obviously this is not real security as nd\_obj.\_password would still show the cleartext password).  
  
Create a setter property for the password attribute. The setter should verify that you are not setting the password to the exact same value as the previous password. If the passwords are identical a ValueError exception should be raised. If the password is new, then your code should update the private password attribute.  
  
  
5. Recreate your Interface class from Exercise2 except this time use an @property and a setter for both intf\_mode and for access\_vlan.  
  
For both of these attributes you should use a private attribute i.e. self.\_intf\_mode and self.\_access\_vlan.  
  
For the @property getter (for both attributes) just have the getter return the private attribute.  
  
For the intf\_mode setter, you should check that the new value is either "access" or "trunk". If any other value is used, raise an exception. If intf\_mode is set to trunk, then you set the access\_vlan attribute to None.  
  
For the access\_vlan setter, you should verify that the VLAN ID is an integer (if intf\_mode is "access"). If the VLAN ID is not an integer, then you should raise a ValueError exception. If intf\_mode is trunk, then you should always set the access\_vlan value to None. In other words, any "intf\_obj.access\_vlan = value" operation should result in "intf\_obj.access\_vlan = None" (if the interface is trunking).  
  
  
  
  
**CLASS OUTLINE**

1. **Classes and Objects Overview**
   1. Why use classes and objects? [0:29]
   2. Situations where classes and objects make sense [0:42]
      1. Instead of repeatedly passing shared data [1:55]
   3. Objects - containers for shared data [2:12]
   4. Methods - functions that can operate on shared data [2:24]
   5. Classes - Blueprint for how objects are created and what objects can do [2:54]
2. **Dunder-init and Creating Objects**
   1. Class Syntax [0:30]
      1. Class keyword [0:33]
      2. Class name [0:35]
         1. Can inherit from other classes, default is to inherit from base ‘object’ [1:05]
   2. Method Syntax - similar to syntax for functions [1:45]
      1. ‘def’ keyword and a set of parameters [1:50]
      2. Dunder-init() method - \_\_init\_\_ (two underscores, the word ‘init’ two trailing underscore) [2:00]
         1. ‘self’ - first parameter [2:17]
   3. Dunder-init() - The Initialization Method [3:00]
      1. \_\_init\_\_ arguments - same rules as functions except for “self” [3:06]
      2. Bind arguments to the object [3:23]
         1. Attributes of the object [4:22]
         2. Essentially variables stuck to the object [5:10]
   4. Classes and Objects - Creating Objects [6:11]
      1. You do not need to pass in “self” this will get passed in automatically [7:10]
      2. You can pass in arguments using position, name, or a mix of the two [7:20]
   5. What about ‘self’? [7:30]
      1. Dunder-new creates an object in memory - "self" is a reference to this object [7:57]
   6. Creating an Object - self and the object variable name both refer to the same thing [9:10]
   7. Multiple objects can exist at the same time [10:44]
3. **Methods**
   1. Methods [0:51]
      1. \_\_init\_\_() [1:05]
      2. write() [1:18]
      3. read() [1:45]
      4. login() [2:04]
   2. Methods Syntax [2:20]
      1. “def” keyword [2:21]
      2. Method name [2:27]
      3. Parameters, with the “self” parameter first [2:40]
   3. Using Methods [3:17]
      1. You can call a sequence of methods [4:09]
      2. Example - sequence of read and write calls [5:08]
      3. Example - login method [6:04]
4. **Static Methods**
   1. Options [0:30]
   2. What are static methods? [1:42]
      1. A static method is a method inside a class that does not use “self” and uses a special decorator "@staticmethod" [2:00]
         1. Decorators: special modifiers of functions or methods [2:37]
   3. Arguments in favor of static methods over normal method [3:29]
      1. Cleaner code [3:51]
      2. Marginally less resources intensive [4:13]
   4. Arguments in favor of static methods over functions [4:55]
      1. Want to be able to use class hierarchies [5:03]
      2. Cleaner code [7:10]
   5. Example from the Netmiko library [7:47]
5. **Special Names**
   1. “Private” Attributes - (single leading underscore) [0:13]
   2. Name Mangling - (double leading underscore only) [1:45]
   3. Magic Methods (special methods, dunder methods) [4:01]
6. **Properties**
   1. We want an action to happen when we retrieve or change an attribute [1:15]
      1. One solution “getters” and “setters” [1:25]
         1. Potentially requires a lot of boilerplate code [3:05]
         2. Would break existing code that relied on previously created attributes [3:20]
      2. Alternate solution @property decorator - allows you to convert data attributes to getter/setter methods while retaining the same programmatic interface [5:35]
7. **Properties - Setters**
   1. Format [0:32]
   2. Making the setter more useful [1:30]
8. **Property Example**
   1. Code still has a telnet connection to the previous host [0:30]
      1. Expand setter to delete old connection and then create a new connection to the new host [1:00]
      2. Create a method that creates the new connection and completes login process [1:50]
9. **Deleters**
   1. @host.deleter property that creates a method that deletes the associated attribute [0:20]