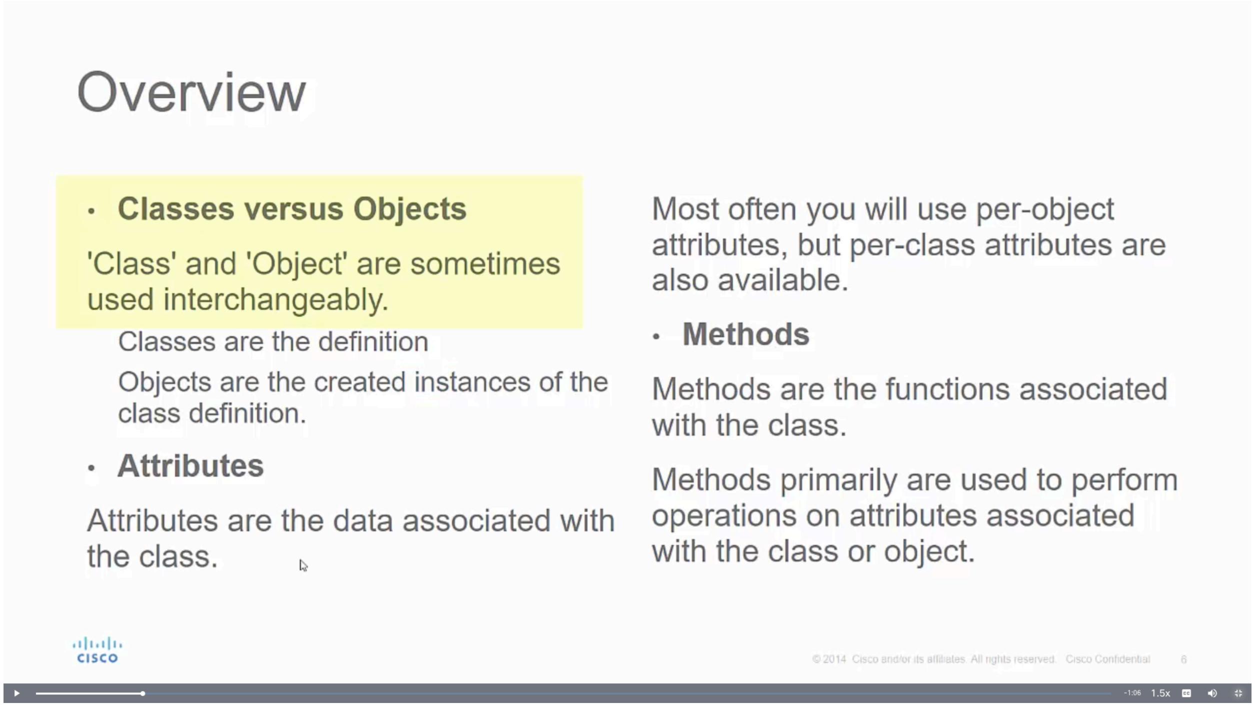

Overview

- **Classes versus Objects**

'Class' and 'Object' are sometimes used interchangeably.

  Classes are the definition

  Objects are the created instances of the class definition.

- **Attributes**

Attributes are the data associated with the class.

Most often you will use per-object attributes, but per-class attributes are also available.

- **Methods**

Methods are the functions associated with the class.

Methods primarily are used to perform operations on attributes associated with the class or object.

One of the first things to understand about OOP is the terminology. The following terms are used for describing object-oriented functionality and data, and must be understood.

- **Classes versus Objects:** In OOP, the formal definition of a particular type of object is called a class. Thus you will define a class in your code, using the 'class' keyword. The class is the template of the actual object that you will later create. There will be one definition the class.

  When you create an instance of that class, that instance is called an object. Creating an instance of an object is called instantiation. When you instantiate an object, you refer to the name of the class, which tells Python what type of object you wish to create. There will be zero, one, or many instances of an object.

  For example, you may define a switch class. Then in your code, you will create some number of switch objects.

- **Attributes:** Attributes are the data associated with instances your class. Every object instance has its own data. Attributes are sometimes called variables, members, or member data. For example, you may define a switch class, and then instantiate some number of switch objects. Each of those objects will have its own data; each object with have its own switch name, software version, ports, and so on.

  There are actually two types of attributes: class attributes and instance (or object) attributes. You will most always use instance variables, which are specific to the object instance in which they reside. However, there is also the lesser-known concept of a class variable, of which there is one per class. For most cases, instance attributes are used.

- **Methods:** Methods are the functions that you will call in order to perform some operation on your object. Operations can be as simple as getting or setting object attributes, or as involved as connecting to a device using some mechanism such as Pexpect or an SDN controller.

  One of the most fundamental ideas of object-oriented programming is the idea that if you call a method on an object, you should not need to know about the details of how the object is going to accomplish the task that you give it. So for example, if your goal is to get the software version of the network device you are connecting to, then you should not have to worry about how the object gets that version information. It may connect via SSH and do a `show version brief`, or there may be another command required on a different device such as a wireless controller or AP. If you have created network device objects to take care of these details, then the rest of my application need not concern itself with how to deal with device type A or device type B.