



# Student Worksheet

## Reading Week 6 Class 1





<b>Subheading 1</b>	
<b>Section 1</b>	Classes provide a means of <b><u>bundling</u></b> data and functionality together. Creating a new class creates a new <i>type</i> of object, allowing new <i>instances</i> of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by its class) for modifying its state.

<b>Section 2</b>	Compared with other programming languages, Python's class mechanism adds classes with a minimum of new syntax and semantics. It is a mixture of the class mechanisms found in C++ and Modula-3. Python classes provide all the standard features of Object Oriented Programming: the class inheritance mechanism allows multiple base classes, a derived class can <b><u>override</u></b> any methods of its base class or classes, and a method can call the method of a base class with the same name. Objects can contain arbitrary amounts and kinds of data. As is true for modules, classes partake of the dynamic nature of Python: they are created at runtime, and can be modified further after creation.
<b>Section 3</b>	In C++ terminology, normally class members (including the data members) are <i>public</i> (except see below <a href="#">Private Variables</a> ), and all member functions are <i>virtual</i> . As in Modula-3, there are no shorthands for referencing the object's members from its methods: the method function is declared with an explicit first argument representing the object, which is provided implicitly by the call. As in Smalltalk, classes themselves are objects. This provides semantics for importing and renaming. Unlike C++ and Modula-3, built-in types can be used as base classes for extension by the user. Also, like in C++, most built-in operators with special syntax (arithmetic operators, subscripting etc.) can be redefined for class instances.





<b>Section 4</b>	Lacking universally accepted terminology to talk about classes, I will make occasional use of Smalltalk and C++ terms. I would use Modula-3 terms, since its object-oriented semantics are closer to those of Python than C++, but I expect that few readers have heard of it.)
<b>Subheading 2</b>	
<b>Section 5</b>	Objects have individuality, and multiple names (in multiple scopes) can be bound to the same object. This is known as aliasing in other languages. This is usually not appreciated on a first glance at Python, and can be safely ignored when dealing with immutable basic types (numbers, strings, tuples). However, aliasing has a possibly surprising effect on the semantics of Python code involving mutable objects such as lists, dictionaries, and most other types.
<b>Section 6</b>	This is usually used to the benefit of the program, since aliases behave like pointers in some respects. For example, passing an object is cheap since only a pointer is passed by the implementation; and if a function modifies an object passed as an argument, the caller will see the change — this eliminates the need for two different argument passing mechanisms as in Pascal.
<b>Subheading 3</b>	
<b>Section 7</b>	<p>Classes introduce a little bit of new syntax, three new object types, and some new semantics.</p> <p>The simplest form of class definition looks like this:</p> <pre>class ClassName:     &lt;statement-1&gt;     .</pre>





```
.  
. .  
<statement-N>
```

Class definitions, like function definitions (`def` statements) must be executed before they have any effect. (You could conceivably place a class definition in a branch of an `if` statement, or inside a function.)

### Skimming

#### 1. Match the subheadings with the sections

- Class Definition Syntax
- Names and Objects
- Classes

### Dictionary

- What is the best meaning for the word “bundling” in Section 1?
  - To sell different items
  - To cause something to move quickly
  - To group together
- What is the best meaning for the word “override” in Section 2?
  - It is done so that a child class can give its own implementation to a method which is already provided by the parent class
  - It occurs when two or more methods in one class have the same method name but different parameters.

#### 4. Reading Comprehension. Read Section 1 carefully and answer the question

- The same object can be connected to different names. T F
- Class definitions have effects before execution. T F
- There is a universally accepted terminology to talk about classes. T F





4. Classes have to be modified during the creation. T F
5. Build in operators allow you to change their definition. T F

