

GLS UNIVERSITY

03I01601 INTRODUCTION TO PYTHON.
UNIT– IV

Data Structure, Classes and Objects

- **Classes and Objects**

-

- Python is an object-oriented programming language.
- Object is simply a collection of data (variables) and methods (functions) that act on those data. And, class is a blueprint for the object.
- Think of class as a sketch (prototype) of a house. It contains all the details about the floors, doors, windows etc. Based on these descriptions we build the house. House is the object.
- An object is also called an instance of a class.

Data Structure, Classes and Objects

- **Defining a Class**

- Like function definitions begin with the keyword `def`, in Python, we define a class using the keyword `class`.
- The name of the class immediately follows the keyword `class` followed by a colon
 - **Syntax:**
 - `class ClassName:`
 - 'Optional class documentation string'
 - `class_suite`
- A class creates a new local namespace where all its attributes are defined. Attributes may be data or functions.
- There are also special attributes in it that begins with double underscores (`__`). For example, `__doc__` gives us the docstring of that class.

Data Structure, Classes and Objects

- **Attributes and Methods in class:**
- Functionalities in class are defined by setting attributes, which act as containers for data and functions related to those attributes. Those functions are called methods. When user define methods, it is always required to provide the first argument to the method with a **self** keyword.
 - Syntax: `def method(self, <parameter>):`
 -
- **Creating an Object**
- To create instances of a class, user can call the class using class name. It is similar to a function call.
- Syntax:
 - `Obj = classname()`
- **Accessing Attributes**
- User can access the object's attributes using the dot operator with object. Class variable would be accessed using class name.

Data Structure, Classes and Objects

- **Constructors in Python**

-

- A constructor is a class function that begins with double underscore (`__`). The name of the constructor is always the same `__init__()`.
- This special function gets called whenever a new object of that class is instantiated.
- While creating an object, a constructor can accept arguments if necessary. When you create a class without a constructor, Python automatically creates a default constructor that doesn't do anything.
- Normally it is use to initialize all the variables.

Data Structure, Classes and Objects

- **Built-In Class Attributes**

-

- Every Python class keeps following built-in attributes and they can be accessed using dot operator like any other attribute –
 - `__dict__`: Dictionary containing the class's namespace.
 - `__doc__`: Class documentation string or none, if undefined.
 - `__name__`: Class name.
 - `__module__`: Module name in which the class is defined. This attribute is "`__main__`" in interactive mode.
 - `__bases__`: A possibly empty tuple containing the base classes, in the order of their occurrence in the base class list.

Data Structure, Classes and Objects

Destroying Objects (Garbage Collection)

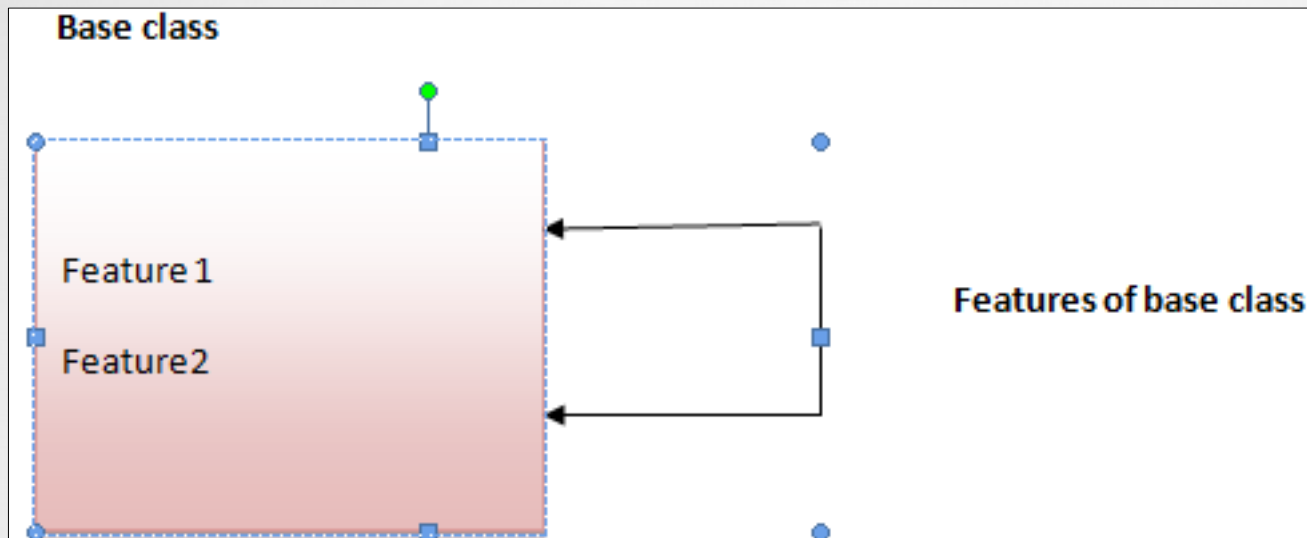
-
- Python deletes unneeded objects (built-in types or class instances) automatically to free the memory space. The process by which Python periodically reclaims blocks of memory that no longer are in use is termed Garbage Collection.
- A class can implement the special method `__del__()`, called a destructor, that is invoked when the instance is about to be destroyed. This method might be used to clean up any non memory resources used by an instance.

Data Structure, Classes and Objects

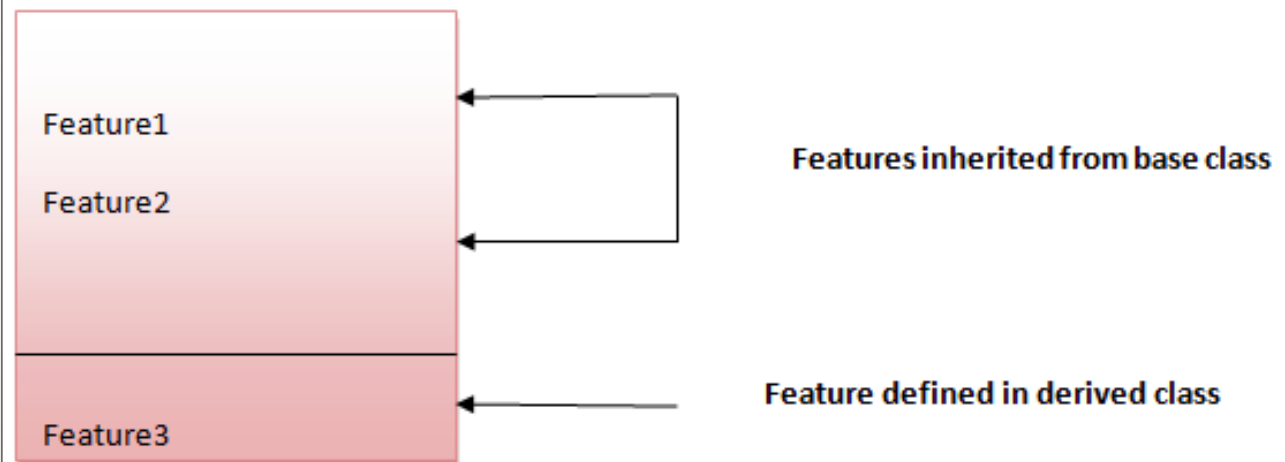
Inheritance

-
- Inheritance is used to specify that one class will get most or all of its features from its parent class. The new class is called child class or derived class and the main class from which it inherits the properties is called base class or parent class.
- The child class or derived class inherits the features from the parent class, adding new features to it.
- It facilitates re-usability of code.

Data Structure, Classes and Objects



Derived class: (Inherited from base class)



Data Structure, Classes and Objects

Inheritance

-
- **Syntax:**
 - `class SubClassName (ParentClass1[, ParentClass2, ...]):`
 - 'Optional class documentation string'
 - `class_suite`
- **Syntax:**
 - `class BaseClass:`
 - Body of base class
 - `class DerivedClass(BaseClass):`
 - Body of derived class
-
- The name `BaseClassName` must be defined in a scope containing the derived class definition.