EdYoda Digital University

Python-21 March 2022

Batch-DS250322

Sagar Sarkar

Day 28-10 May OOPs-2

- Constructors
- Inheritence
- Polymorphism
- Method Overriding
- Access Specifiers

Constructors

- Constructors are generally used for instantiating an object.
- The task of constructors is to initialize(assign values) to the data members of the class when an object of the class is created.
- In Python the __init__() method is called the constructor and is always called when an object is created.
  Syntax of constructor declaration :
- `def __init__(self):`
- `    # body of the constructor`

Types of constructors :

- **default constructor:** The default constructor is a simple constructor which doesn’t accept any arguments. Its definition has only one argument which is a reference to the instance being constructed.
- **parameterized constructor:** constructor with parameters is known as parameterized constructor. The parameterized constructor takes its first argument as a reference to the instance being constructed known as self and the rest of the arguments are provided by the programmer.

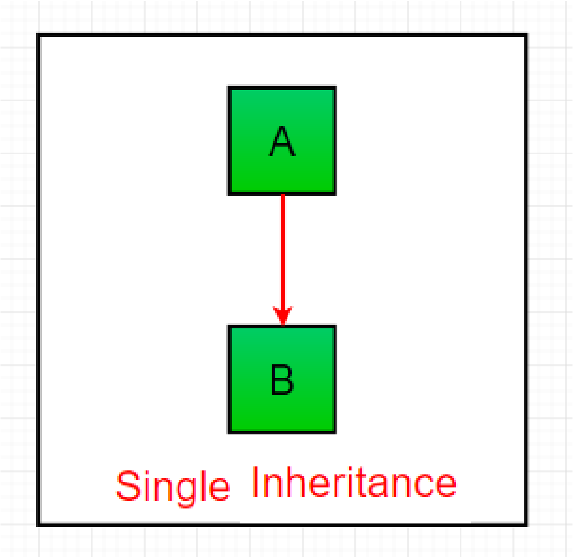
Constructors

Types of constructors :

- **default constructor:** The default constructor is a simple constructor which doesn’t accept any arguments. Its definition has only one argument which is a reference to the instance being constructed.
- **parameterized constructor:** constructor with parameters is known as parameterized constructor. The parameterized constructor takes its first argument as a reference to the instance being constructed known as self and the rest of the arguments are provided by the programmer.

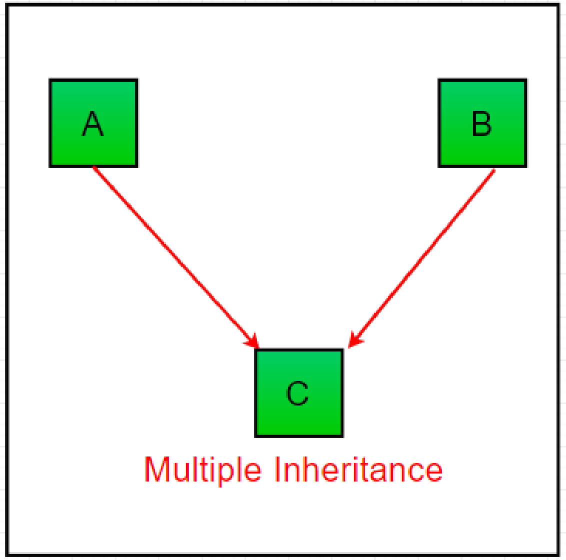
Inheritence

Single Inheritance: Single inheritance enables a derived class to inherit properties from a single parent class, thus enabling code reusability and the addition of new features to existing code.



Inheritence

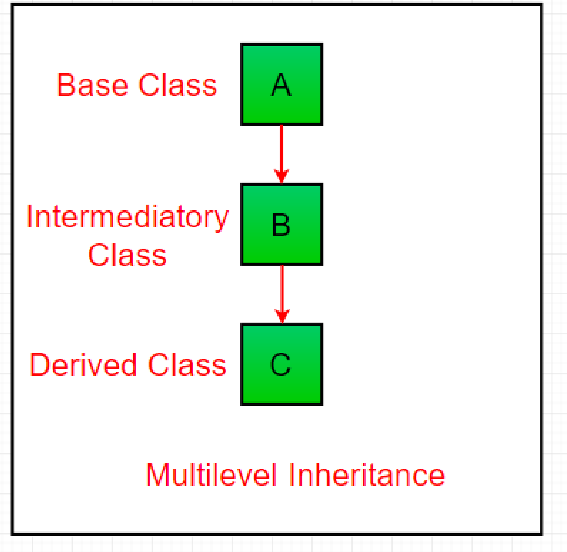
Multiple Inheritance: When a class can be derived from more than one base class this type of inheritance is called multiple inheritance. In multiple inheritance, all the features of the base classes are inherited into the derived class.



Inheritence

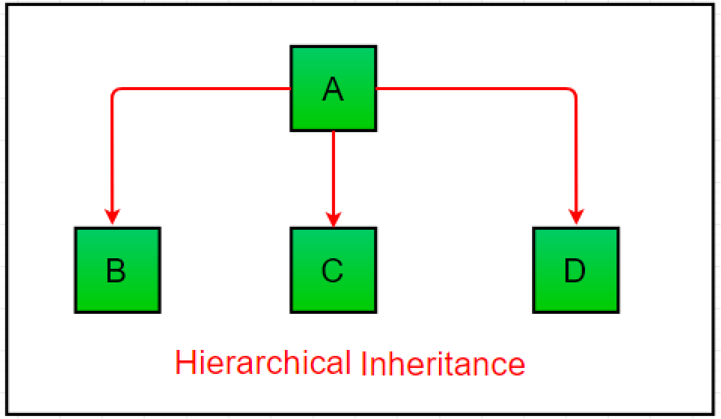
Multilevel Inheritance

In multilevel inheritance, features of the base class and the derived class are further inherited into the new derived class. This is similar to a relationship representing a child and grandfather.



Inheritence

Hierarchical Inheritance: When more than one derived classes are created from a single base this type of inheritance is called hierarchical inheritance. In this program, we have a parent (base) class and two child (derived) classes.

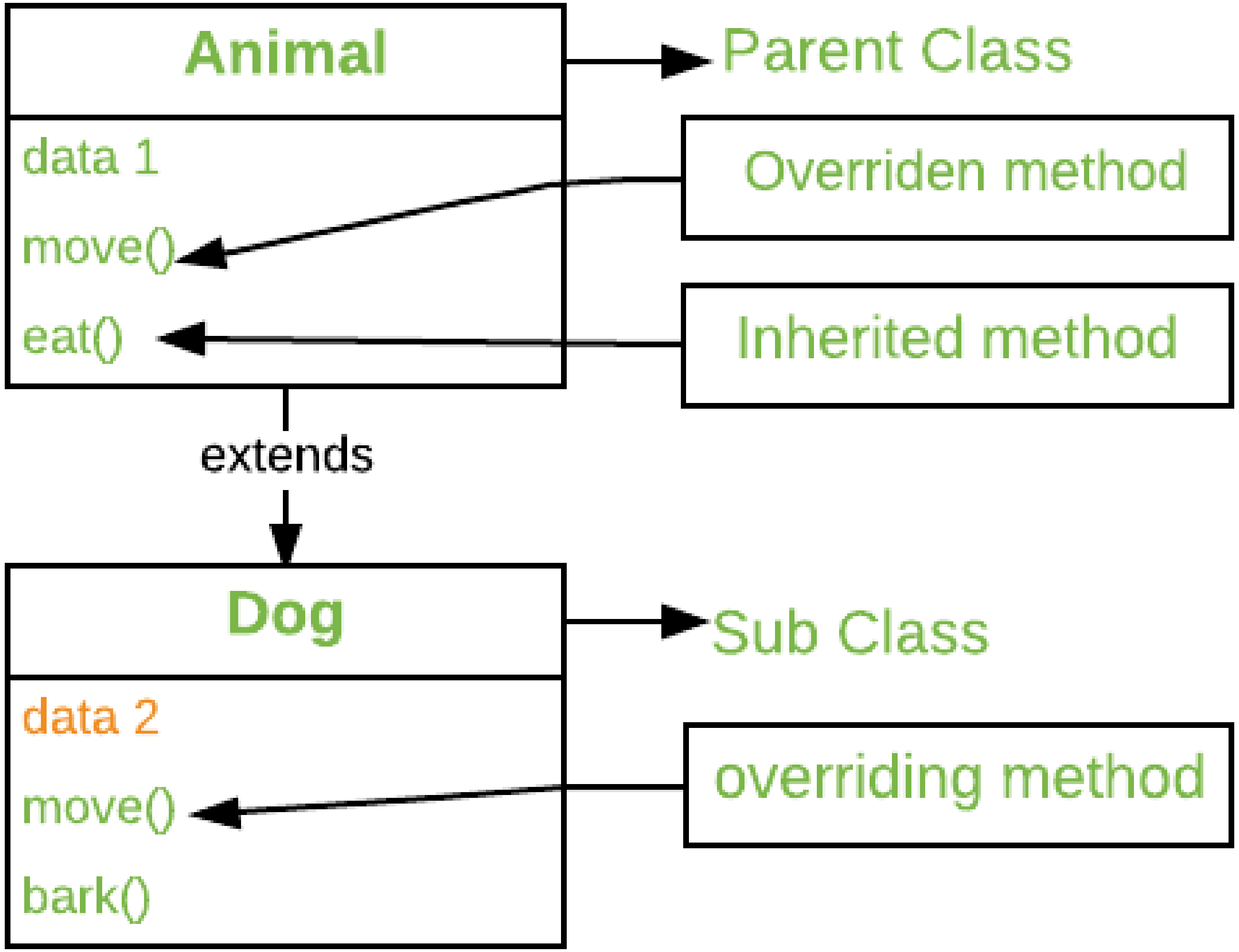


Hierarchical Inheritance

Method Overriding

Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to **override** the method in the super-class.

Method Overriding



Animal

data 1

move()

eat()

Parent Class

Overriden method

Inherited method

extends

Dog

data 2

move()

bark()

Sub Class

overriding method

Method Overriding

he version of a method that is executed will be determined by the object that is used to invoke it.

If an object of a parent class is used to invoke the method, then the version in the parent class will be executed, but if an object of the subclass is used to invoke the method, then the version in the child class will be executed.

In other words, it is the type of the object being referred to (not the type of the reference variable) that determines which version of an overridden method will be executed

Access Modifiers in Python

A Class in Python has three types of access modifiers:

- **Public Access Modifier**
- **Protected Access Modifier**
- **Private Access Modifier**

Access Modifiers in Python

Public Access Modifier:

- The members of a class that are declared public are easily accessible from any part of the program.
- All data members and member functions of a class are public by default

Access Modifiers in Python

Protected Access Modifier:

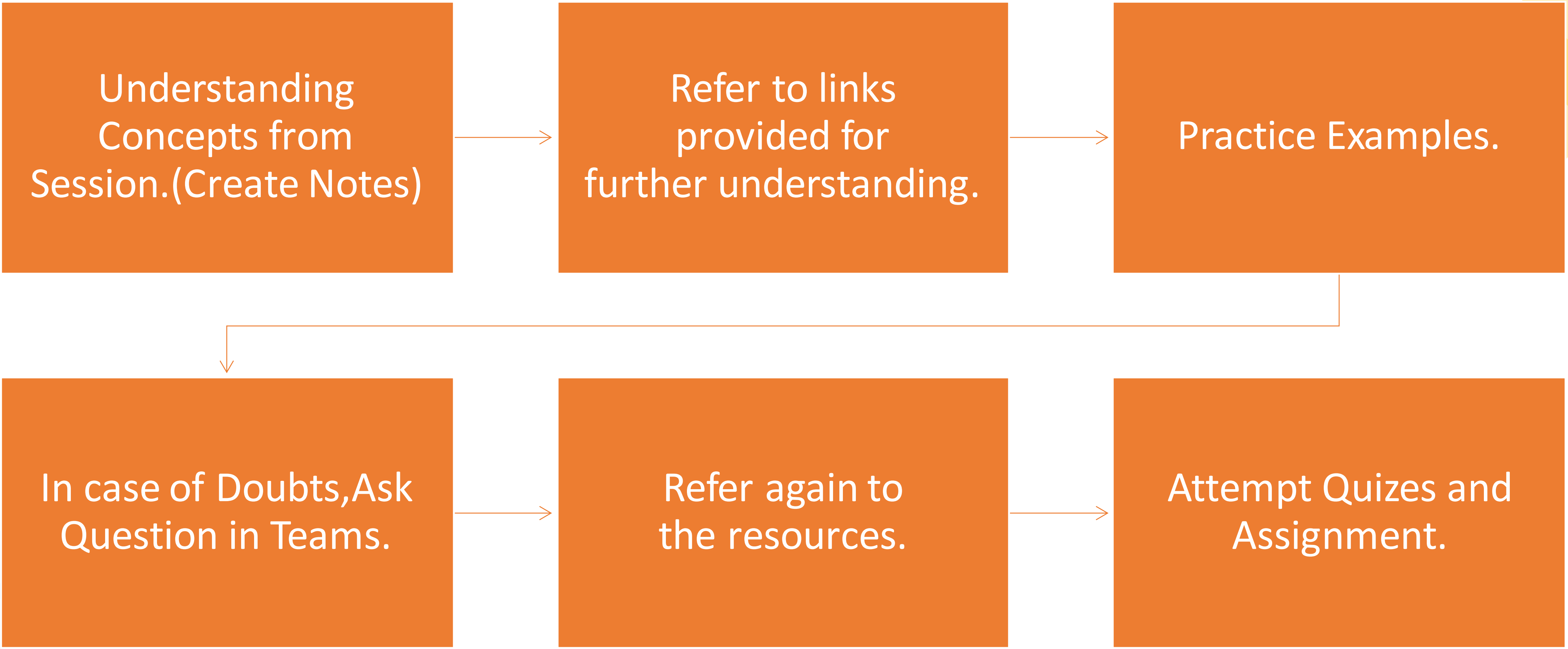
- The members of a class that are declared protected are only accessible to a class derived from it.
- Data members of a class are declared protected by adding a single underscore ‘_’ symbol before the data member of that class.

Access Modifiers in Python

Private Access Modifier:

- The members of a class that are declared private are accessible within the class only, private access modifier is the most secure access modifier.
- Data members of a class are declared private by adding a double underscore ‘__’ symbol before the data member of that class.

Approach to learning Python



Understanding Concepts from Session.(Create Notes)

Refer to links provided for further understanding.

Practice Examples.

In case of Doubts,Ask Question in Teams.

Refer again to the resources.

Attempt Quizes and Assignment.

Anyone ??