EdYoda Digital University

Python-21 March 2022

Batch-DS250322

Sagar Sarkar

Day 29-11 May OOPs-3

- ABC
- Static and class methods
- Dunder and Magic Methods
- MRO

Class method vs Static method

- The @classmethod decorator is a built-in function decorator that is an expression that gets evaluated after your function is defined. The result of that evaluation shadows your function definition.
  A class method receives the class as an implicit first argument, just like an instance method receives the instance
  Syntax:

```
• class C(object):
•     @classmethod
•     def fun(cls, arg1, arg2, ...):
•        ....
```

- **fun:** function that needs to be converted into a class method
- **returns:** a class method for function.

Class method vs Static method

- A class method is a method that is bound to the class and not the object of the class.
- They have the access to the state of the class as it takes a class parameter that points to the class and not the object instance.
- It can modify a class state that would apply across all the instances of the class. For example, it can modify a class variable that will be applicable to all the instances

Class method vs Static method

Static Method

- A static method does not receive an implicit first argument.
- **Syntax:**

```
class C(object):
    @staticmethod
    def fun(arg1, arg2, ...):
        ...
```

- **returns:** a static method for function fun.
- A static method is also a method that is bound to the class and not the object of the class.
- A static method can’t access or modify the class state.
- It is present in a class because it makes sense for the method to be present in class.

Class method vs Static method

Class method vs Static Method

- A class method takes cls as the first parameter while a static method needs no specific parameters.
- A class method can access or modify the class state while a static method can't access or modify it.
- In general, static methods know nothing about the class state. They are utility-type methods that take some parameters and work upon those parameters. On the other hand class methods must have class as a parameter.
- We use @classmethod decorator in python to create a class method and we use @staticmethod decorator to create a static method in python.

When to use what?

- We generally use class method to create factory methods. Factory methods return class objects (similar to a constructor) for different use cases.
- We generally use static methods to create utility functions.

ABC

An abstract class can be considered as a blueprint for other classes. It allows you to create a set of methods that must be created within any child classes built from the abstract class.

A class which contains one or more abstract methods is called an abstract class.

An abstract method is a method that has a declaration but does not have an implementation.

While we are designing large functional units we use an abstract class.

When we want to provide a common interface for different implementations of a component, we use an abstract class.

ABC

Why use Abstract Base Classes :

- By defining an abstract base class, you can define a common Application Program Interface(API) for a set of subclasses.
- This capability is especially useful in situations where a third-party is going to provide implementations, such as with plugins, but can also help you when working in a large team or with a large code-base where keeping all classes in your mind is difficult or not possible.

How Abstract Base classes work :

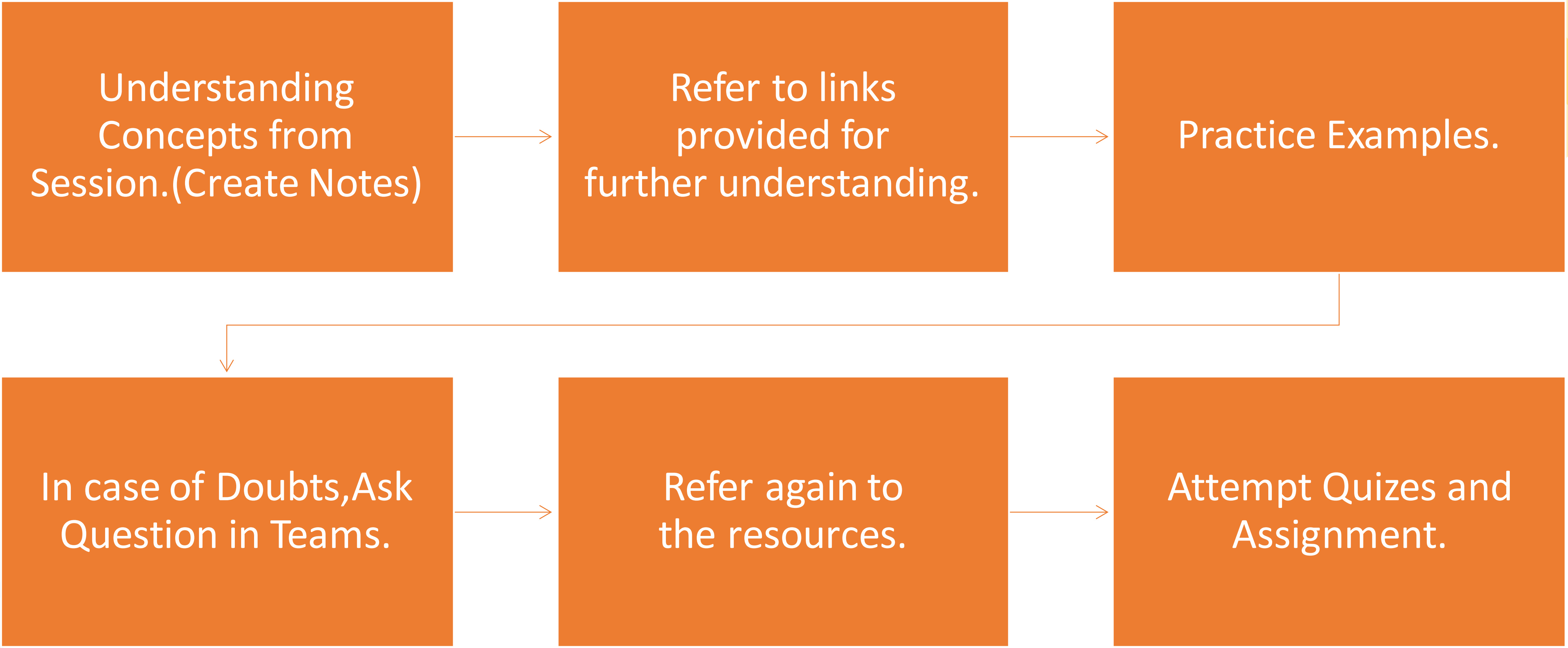
- By default, Python does not provide abstract classes. Python comes with a module that provides the base for defining Abstract Base classes(ABC) and that module name is ABC.
- ***ABC*** works by decorating methods of the base class as abstract and then registering concrete classes as implementations of the abstract base. A method becomes abstract when decorated with the keyword @abstractmethod

ABC

Concrete Methods in Abstract Base Classes : Concrete classes contain only concrete (normal)methods whereas abstract classes may contain both concrete methods and abstract methods.

The concrete class provides an implementation of abstract methods, the abstract base class can also provide an implementation by invoking the methods via super().

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 ??