***ABSTRACT CLASS:***

1.Abstraction is the concept in object-oriented programming that is used to hide the internal functionality of the classes from the users

2.Abstraction is implemented using the abstract classes.

3.A class containing more than one abstract method is called an abstract class.

4. abstract method is one that doesn't have any implementation.

5.real-life example: when we press a button on a TV remote to change the channel, we don’t know how it does that, we are just interested in that when we press a particular button, it changes the channel.

6.We cannot create an abstract class in Python directly. However, Python does provide a module that allows us to define abstract classes. The module we can use to create an abstract class in Python is abc(abstract base class) module.

Syntax:

*from abc import ABC*

*class <Abstract\_Class\_Name>(ABC):*

*# body of the class*

Example 1:

*from abc import ABC, abstractmethod*

*class DemoAbstractClass(ABC):*

*@abstractmethod*

*def abstract\_method\_name(self):*

*Pass*

Example 2:

*from abc import ABC, abstractmethod*

*class Shape(ABC):*

*def \_\_init\_\_(self, shape\_name):*

*self.shape\_name = shape\_name*

*@abstractmethod*

*def draw(self):*

*pass*

*class Circle(Shape):*

*def \_\_init\_\_(self):*

*super().\_\_init\_\_("circle")*

*def draw(self):*

*print("Drawing a Circle")*

*circle = Circle()*

*circle.draw()*