Single Inheritance In Python:

Single inheritance in Python refers to the concept where a class can inherit attributes and methods from only one parent class. In other words, a subclass can have only one superclass.

Here's a simple example demonstrating single inheritance:

CODE:

```
class Animal:
    def   init   (self, name):
        self.name = name

    def speak(self):
        pass

class Dog(Animal):
    def speak(self):
        return "Woof!"

# Creating an instance of the Dog class
dog = Dog("Buddy")
print(dog.name)  # Output: Buddy
print(dog.speak())  # Output: Woof!
```

Output:

```
C:\Users\saksh\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\saksh\PycharmProjects\pythonProject\35.py
Buddy
Woof!
```

In this example, the `Animal` class is the superclass, and the `Dog` class is the subclass that inherits from `Animal`. The `Dog` class overrides the `speak` method of the `Animal` class with its own implementation.

Single inheritance promotes simplicity and clarity in the class hierarchy. However, Python also supports multiple inheritance, where a class can inherit from multiple parent classes.

Multiple Inheritance in Python:

Multiple inheritance in Python refers to the concept where a class can inherit attributes and methods from more than one parent class. This allows for more complex class hierarchies where a subclass can inherit behavior from multiple superclasses.

Here's an example demonstrating multiple inheritance:

Code:

```
class Animal:
    def make_sound(self):
        pass

class Dog(Animal):
    def make_sound(self):
        return "Woof!"

class Cat(Animal):
    def make_sound(self):
        return "Meow!"

class DogCat(Dog, Cat):
    pass

# Creating an instance of the DogCat class
pet = DogCat()
print(pet.make_sound())   # Output: Woof!
```

Output:

```
C:\Users\saksh\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\saksh\PycharmProjects\pythonProject\35_2.py
Woof!
```

Multilevel Inheritance in Python:

Multilevel inheritance in Python occurs when a class inherits from another class, and then that subclass is inherited by another class, forming a hierarchy of inheritance with multiple levels. Each subclass inherits attributes and methods from its immediate parent class and all the way up the chain to the topmost superclass.

Here's an example demonstrating multilevel inheritance:

Code:

```
class Animal:
    def speak(self):
        return "Generic animal sound"

class Dog(Animal):
    def speak(self):
        return "Woof!"

class GermanShepherd(Dog):
    def speak(self):
        return "Woof! I'm a German Shepherd."

# Creating an instance of the GermanShepherd class
dog = GermanShepherd()
print(dog.speak())  # Output: Woof! I'm a German Shepherd.
```

Output:

```
C:\Users\saksh\PycharmProjects\pythonProject\.venv\Scripts\python.exe C:\Users\saksh\PycharmProjects\pythonProject\35_3.py
Woof! I'm a German Shepherd.
```