Learning Assignment:

Inheritance:

The method of inheriting the properties of parent class into a child class is known as inheritance.

Benefits of Inheritance:

1. Code reusability- we do not have to write the same code again and again, we can just inherit the properties we need in a child class.
2. It represents a real world relationship between parent class and child class.
3. It is transitive in nature. If a child class inherits properties from a parent class, then all other sub-classes of the child class will also inherit the properties of the parent class.

Example:

Here is the simple example of inheritance:

Code:

```
class parent:
    def first(self):
        print("This is first function")

class child(parent):
    def second(self):
        print("This is second function")

ob = child()
ob.first()
ob.second()
```

Output:

```
In [3]: runfile('C:/Users/LENOVO/Desktop/Learning Assignment/Example of Inheritence.py',
wdir='C:/Users/LENOVO/Desktop/Learning Assignment')
This is first function
This is second function
```

Types of Inheritance:

1. Single
2. Multiple
3. Multilevel
4. Hierarchical

1. Single Inheritance:

When a child class inherits only a single parent class.

Example:

```
class parent:
    def first(self):
        print("Single Inheritance:")

class child(parent):
    def second(self):
        print("This is an example of single inheritance")

ob = child()
ob.first()
ob.second()
```

Output:

```
In [1]: runfile('C:/Users/LENOVO/Desktop/Learning Assignment/Single Inheritance.py', wdir='C:/
Users/LENOVO/Desktop/Learning Assignment')
Single Inheritance:
This is an example of single inheritance
```

2. Multiple Inheritance:

When a child class inherits from more than one parent class.

Example:

```
class Parent:
    def func1(self):
        print("This is function 1")

class Parent2:
    def func2(self):
        print("This is function 2")

class child(Parent, Parent2):
    def func3(self):
        print("This is function 3")

ob = child()
ob.func1()
ob.func2()
ob.func3()
```

Output:

```
In [4]: runfile('C:/Users/LENOVO/Desktop/Learning Assignment/Multiple Inheritance.py', wdir='C:/
Users/LENOVO/Desktop/Learning Assignment')
This is function 1
This is function 2
This is function 3
```

3. Multilevel Inheritance:

When a child class becomes a parent class for another child class.

Example:

```
class Parent:
    def func1(self):
        print("This is function 1")

class Child(Parent):
    def func2(self):
        print("This is function 2")

class Child2(Child):
    def func3(self):
        print("This is function 3")

ob = Child2()
ob.func1()
ob.func2()
ob.func3()
```

Output:

```
In [17]: runfile('C:/Users/LENOVO/Desktop/Learning Assignment/Multilevel Inheritance.py',
wdir='C:/Users/LENOVO/Desktop/Learning Assignment')
This is function 1
This is function 2
This is function 3
```

4. Hierarchical Inheritance:

Hierarchical inheritance involves multiple inheritance from the same base or parent class.

Example:

```
class Parent:
    def func1(self):
        print("This is function 1")

class Child(Parent):
    def func2(self):
        print("This is function 2")

class Child2(Parent):
    def func3(self):
        print("This is function 3")

ob = Child()
ob1 = Child2()
ob.func1()
ob.func2()
ob1.func3()
```

Output:

```
In [21]: runfile('C:/Users/LENOVO/Desktop/Learning Assignment/Hierarchical Inheritance.py',
wdir='C:/Users/LENOVO/Desktop/Learning Assignment')
This is function 1
This is function 2
This is function 3
```