

Polymorphism

Polymorphism is a word that came from two greek words, poly means many and morphos means forms.

If a variable, object or method perform different behavior according to situation, it is called polymorphism.

- Duck Typing
- Operator Overloading
- Method Overloading
- Method Overriding

Duck Typing

In Python, we follow a principle - If 'it walks like a duck and talks like a duck, it must be a duck' which means python doesn't care about which class of object it is, if it is an object and required behavior is present for that object then it will work. The type of object is distinguished only at runtime. This is called as duck typing.

Duck Typing

Python doesn't care about which class of object it is, in order to call an existing method on an object. If the method is defined on the object, then it will be called.



walk - thapak thapak



walk – tabdak tabdak

Strong Typing

We can check whether the object passed to the method has the method being invoked or not.

hasattr () Function is used to check whether the object has a method or not.

Syntax:- hasattr(object, attribute)

Where attribute can be a method or variable. If it is found in the object then this method returns True else False.

Method Overloading

When more than one method with the same name is defined in the same class, it is known as method overloading.

In python, If a method is written such that it can perform more than one task, it is called method overloading.

Method Overriding

If we write method in the both classes, parent class and child class then the parent class's method is not available to the child class.

In this case only child class's method is accessible which means child class's method is replacing parent class's method.

Method overriding is used when programmer want to modify the existing behavior of a Method.

Method Overriding

```
class Add:
```

```
    def result(self, a, b):
```

```
        print("Addition:", a+b)
```

```
class Multi(Add):
```

```
    def result(self, a, b):
```

```
        print("Multiplication:", a*b)
```

```
m = Multi()
```

```
m.result(10, 20)
```

Method with super() Method

If we write method in the both classes, parent class and child class then the parent class's method is not available to the child class.

In this case only child class's method is accessible which means child class's method is replacing parent class's method.

super () method is used to call parent class's *constructor* or *methods* from the child class.

Syntax:- `super().methodName()`

Operator Overloading

If any operator performs additional actions other than what it is meant for, it is called operator overloading.