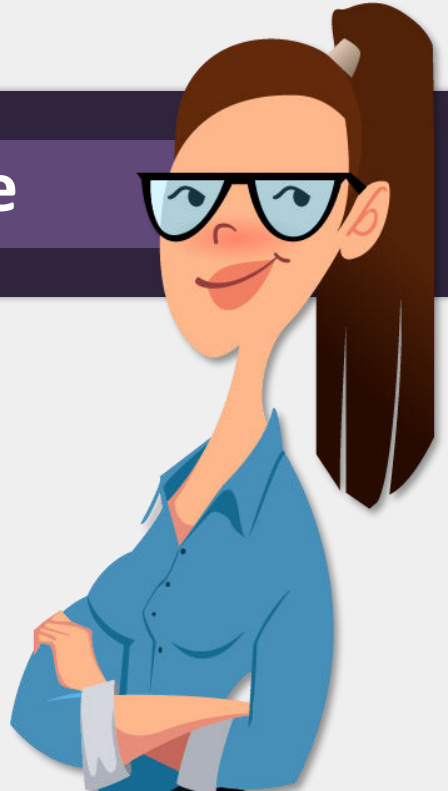
IntelliPaat

Python Certification Course



Introduction to OOPS

IntelliPaat

Introduction To OOP

What is OOP?

O: Object

O: Oriented

P: Programming

IntelliPaat

Introduction To OOP

What is OOP?

For Eg:

Attribute: Name, Age,Color

Behaviour: Singing, Dancing

Parrot

IntelliPaat

Introduction To OOP

What is OOP?

Basic Principle of OOPS:

- Class
- Object
- Inheritance
- Encapsulation
- Polymorphism

IntelliPaat

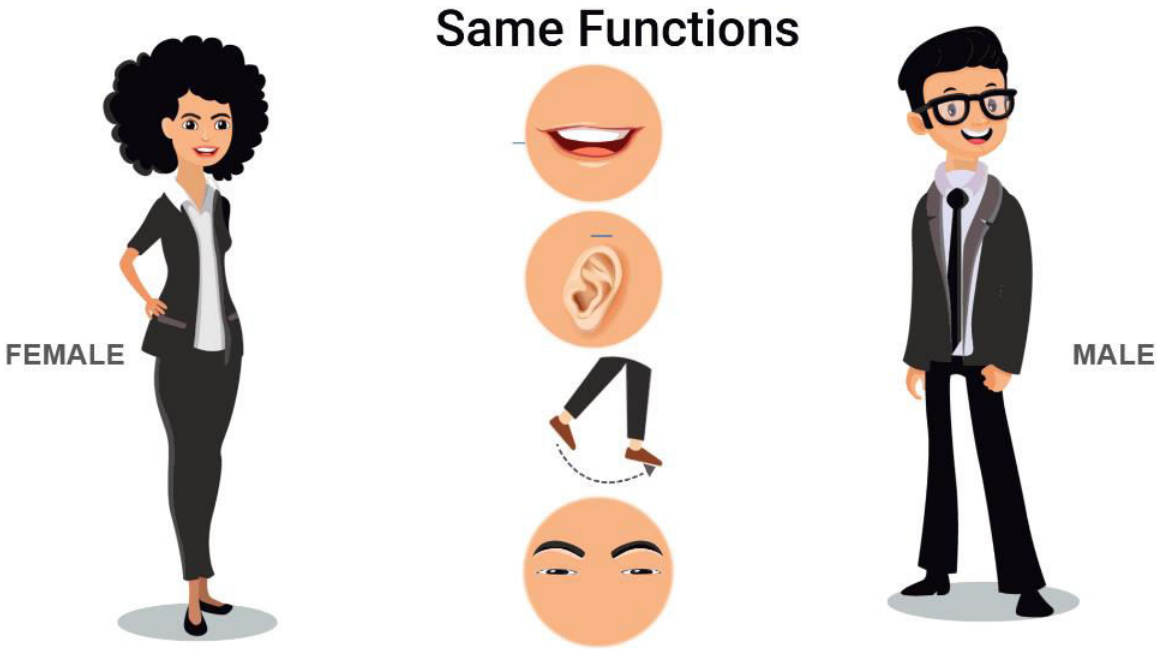
Let’s understand with a Real-Life Example

IntelliPaat

Introduction To OOP

Real-life example of OOP

Every Human Being is Classified into:

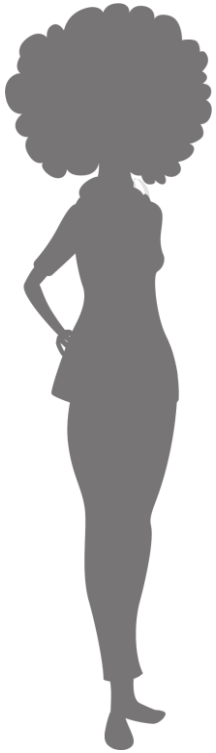


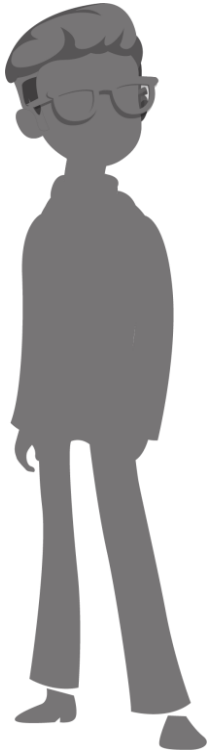
IntelliPaat

Introduction To OOP

Concept of OOP - Class

Consider Human Being is a Class





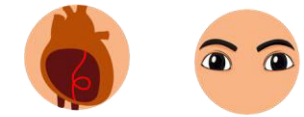
Introduction To OOP

Concept of OOP - Class

Common body features and Functions are Class Attributes

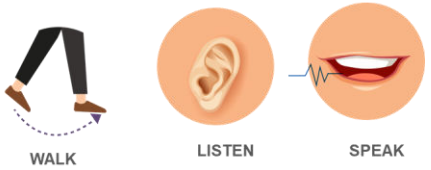
Every Human has:

Common Body Parts:



Common Body Function:

Every Human has:



IntelliPaat

Introduction To OOP

Concept of OOP - Inheritance

Male and Female are inherited from Class Human Being

FEMALE

MALE

IntelliPaat

Introduction To OOP

Concept of OOP - Object

'Name' and 'Age' are object of class MALE

Class: MALE

Name: Victor

Age: 24

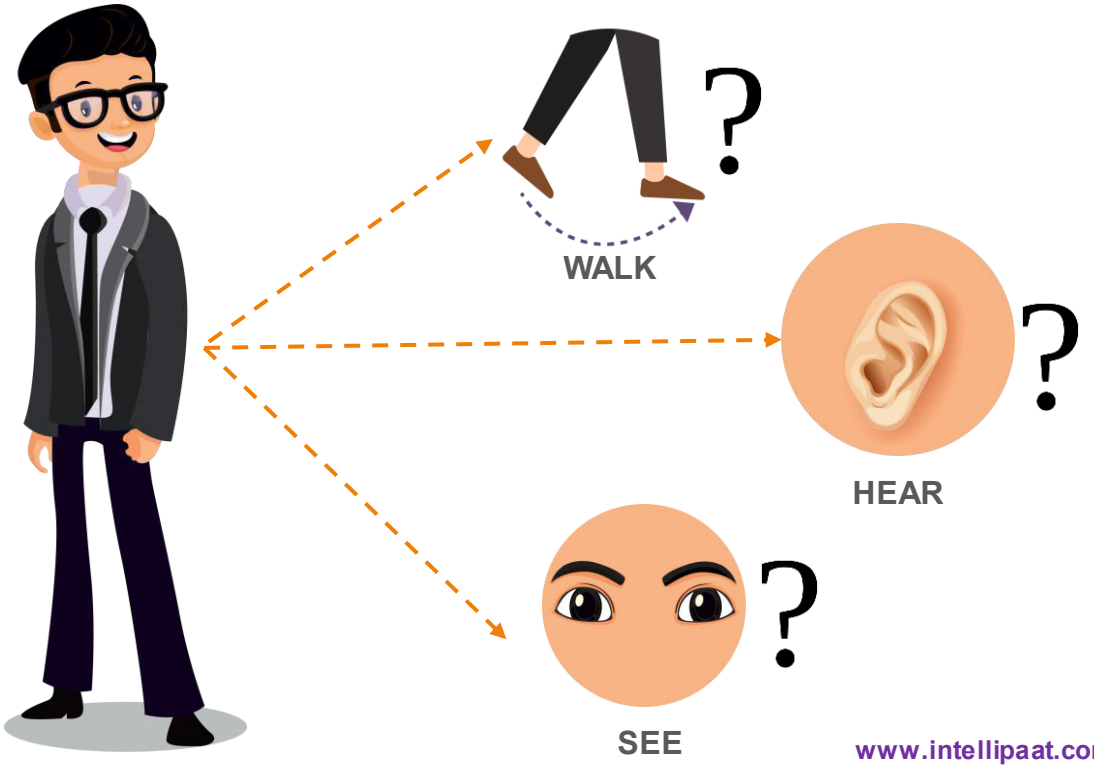
- Objects have a physical existence
- Class is just a logical definition

IntelliPaat

Introduction To OOP

Concept of OOP - Encapsulation

You don't know the detail of how you walk, listen or see. i.e its hidden



IntelliPaat

Introduction To OOP

Concept of OOP - Polymorphism

‘She’ can be a woman, wife, mother and a teacher at the same time

WOMAN WIFE or MOTHER TEACHER

OOPS: Classes and Objects

Things you will learn after this Session

OOPS: Classes and Objects

- What are Classes and Objects?
- How to create a class in Python?
- How to create objects in Python?
- How to access class members?
- Concept of init method/constructor in Python

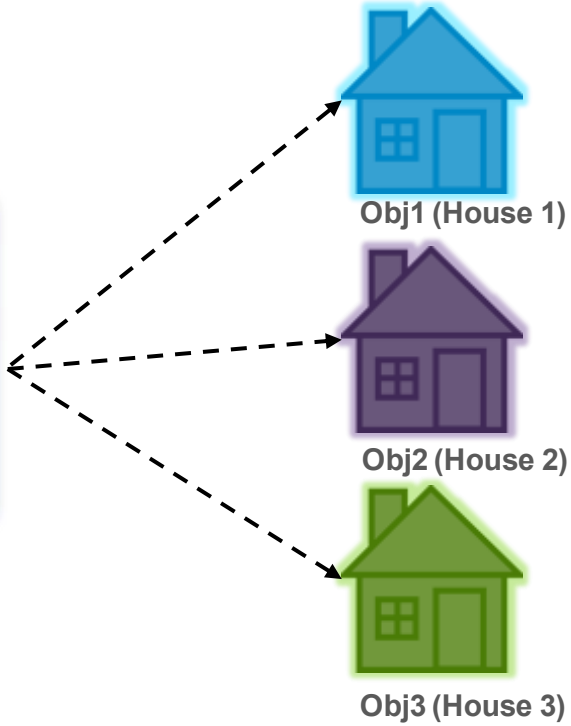
IntelliPaat

Classes and Objects

What are Object and Class?

- Class is a blueprint for an object
- Objects: Defined and created from classes(blueprint)

Class (House Blue Print)



IntelliPaat

Classes and Objects

What are Object and Class?

- Object is the basic unit of object-oriented programming
- An object represents a particular instance of a class
- There can be more than one instance of an object
- Each instance of an object can hold its own relevant data
- Objects with similar properties and methods are grouped together to form a Class

IntelliPaat

Classes and Objects

How to create a Class in Python?

Syntax:

```
class NameOfClass:
        <statement-1>
        .
        .
        <statement-2>
```

Example

```
class ClassName:
        variable= “I am a class
Attribute”
        def function(self):
                print(“I am from
inside the class”)
```

IntelliPaat

Classes and Objects

How to create a Object in Python?

Syntax:

```
<obj-name> = NameOfClass()
```

Example

```
obj1 = ClassName()
```

Here obj1 is an object of class ClassName

IntelliPaat

Classes and Objects

How to access Class Members?

Example

```
obj1 = ClassName()
obj2 = ClassName()
#Creating new instance attribute for obj2
obj2.variable = “I was just created”
print(obj1.variable)
print(obj2.variable)
print(ClassName.variable)
Obj1.function()
```

- Here obj1 and obj2 are object of class ClassName.
- To access the members of a Python class, we use the dot operator.

IntelliPaat

Classes and Objects

`__init__()` method in Python

- __init__ is a special method in Python classes
- Is a constructor method for a class
- __init__ is called when ever an object of the class is constructed

IntelliPaat

Classes and Objects

__init__() method in Python

Example

```
class Student(object):

    def __init__(self, name, branch, year):
            self.name = name
            self.branch = branch
            self.year = year
            print("A student object is created.")

    def print_details(self):
        print("Name:", self.name)
        print("Branch:", self.branch)
        print("Year:", self.year)
```

```
ob1= Student( “Paul","CSE",2019)
ob1.print_details()
```

IntelliPaat

Classes and Objects

__init__() method in Python

Output

```
A student object is created.
Name: Paul
Branch: CSE
Year: 2018
```

IntelliPaat

Demo Test1

T1. Create two new vehicles called car1 and car2. Set car1 to be a red convertible worth $70,000.00 with a name of Ferrari, and car2 to be a blue van named JEEP worth $15,000.00.

Example

```
# define the Vehicle class
class Vehicle:
        name = ""
        kind = "car“
        color = ""
        value = 100.00
        def description(self):
          desc_str = "%s is a %s %s worth $%.2f." %
            (self.name, self.color, self.kind,
            self.value)
           return desc_str
# your code goes here
……..
print(car1.description())
print(car2.description())
```

Demo Test Solution

IntelliPaat

OOPS: Inheritance in Python

Things you will learn after this Session

OOPS: Inheritance in Python

- What is Inheritance?
- Real life example of Inheritance
- Different types of Inheritance in Python
- Overriding vs Overloading

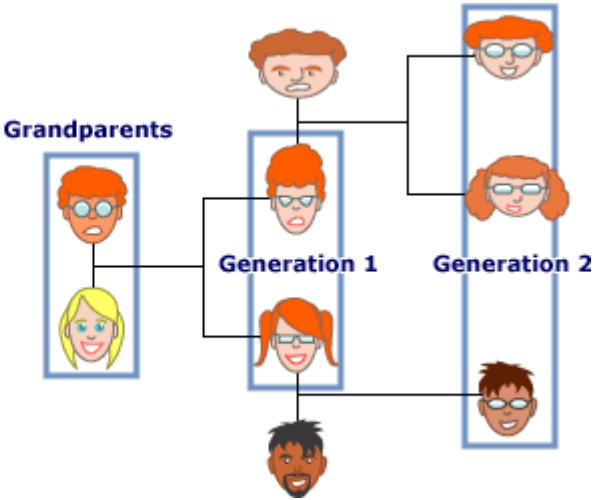
Inheritance in Python

What is Inheritance in Python?

"One class acquiring the property of another class"

Eg: You have inherited qualities from your parents

In a family tree, traits such as hair color and poor eyesight are passed from generation to generation.



IntelliPaat

Inheritance in Python

Inheritance Syntax in Python

Creating a Class in Python

```
class ClassName:
        <statement-1>
        .
        .
        <statement-2>
```

IntelliPaat

Inheritance in Python

Different Types of Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

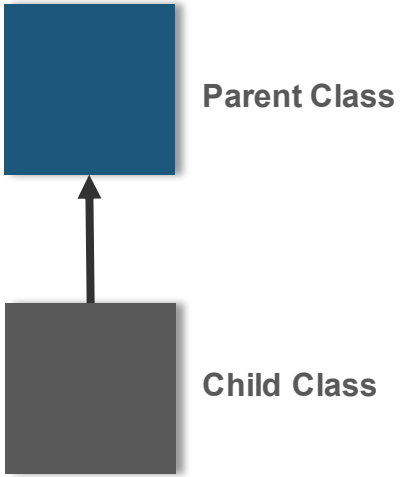
IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Single Inheritance?

"single class inherits from a class"



IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Single Inheritance?

"single class inherits from a class"

Single Inheritance

```
class fruit:
      def __init__(self):
            print("I'm a fruit")
class citrus(fruit):
      def __init__(self):
            super().__init__()
            print("I'm citrus")
Lemon = citrus()
```

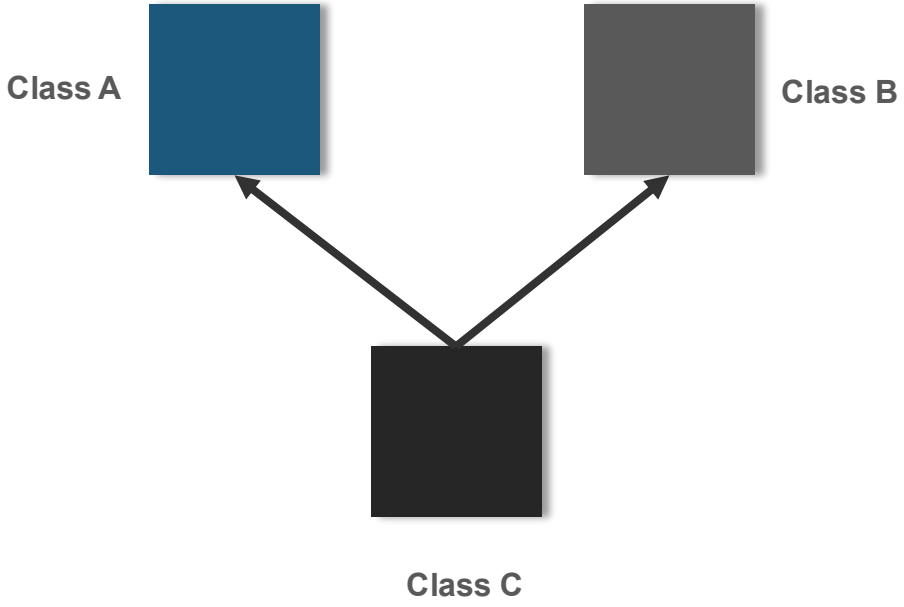
IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Multiple Inheritance?

"A class inherits from multiple class"



IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Multiple Inheritance?

"Multiple class inherits from a class"

Multiple Inheritance

```
class A:
     pass
class B:
     pass
class C(A,B):
     pass
issubclass(C,A) and issubclass(C,B)
```

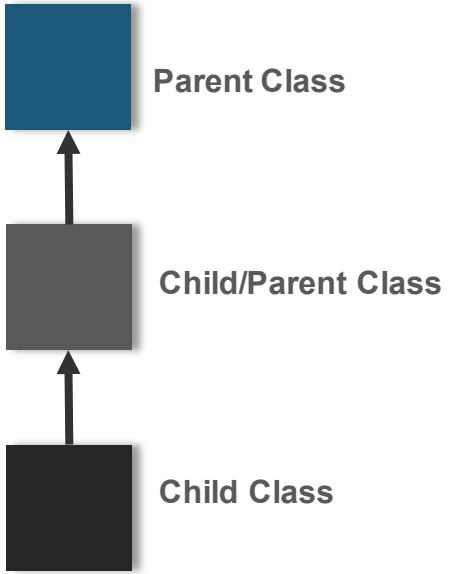
IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Multilevel Inheritance?

"One class inherits from another, which in turn inherits from another"



IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Multilevel Inheritance?

"One class inherits from another, which in turn inherits from another"

Multilevel Inheritance

```
class A:
            x=1
class B(A):
            pass
class C(B):
            pass
cobj=C()
cobj.x
```

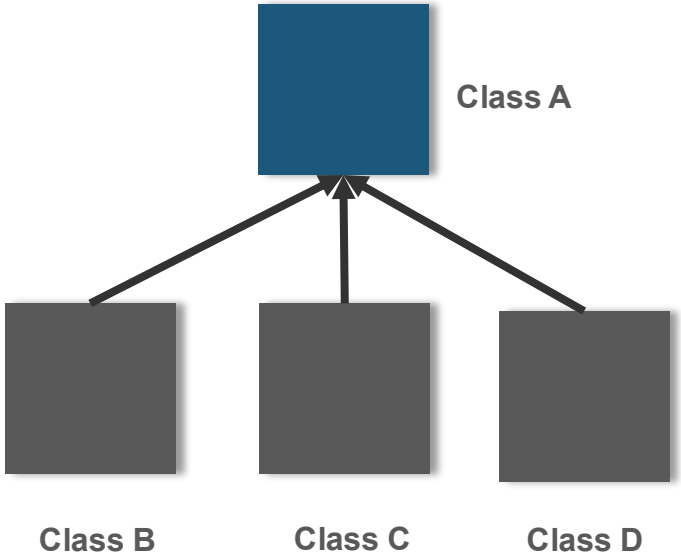
IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Hierarchical Inheritance?

"More than one class inherits from a class"



IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Hierarchical Inheritance?

"More than one class inherits from a class"

Hierarchical Inheritance

```
class A:
          pass
class B(A):
          pass
class C(A):
           pass
issubclass(B,A) and issubclass(C,A)
```

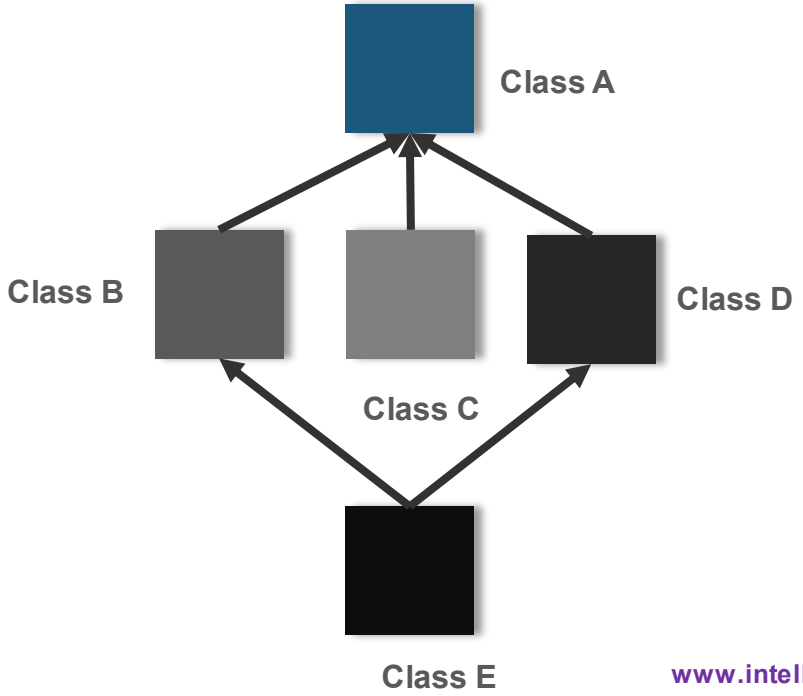
IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Hybrid Inheritance?

"Combination of any two kinds of inheritance"



IntelliPaat

Inheritance in Python

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

What is Hybrid Inheritance?

"Combination of any two kinds of inheritance"

Hybrid Inheritance

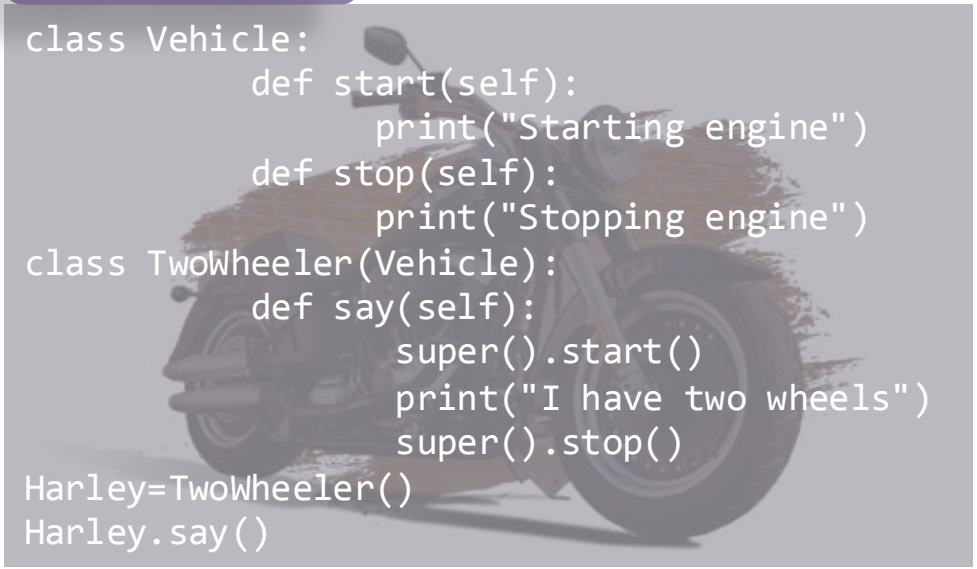
```
>>> class A:
                x=1
>>> class B(A):
                pass
>>> class C(A):
                pass
>>> class D(B,C):
                pass
>>> dobj=D()
>>> dobj.x
```

Inheritance in Python

Inheritance Super Function

"Used to call a method from the parent class"

Super Function



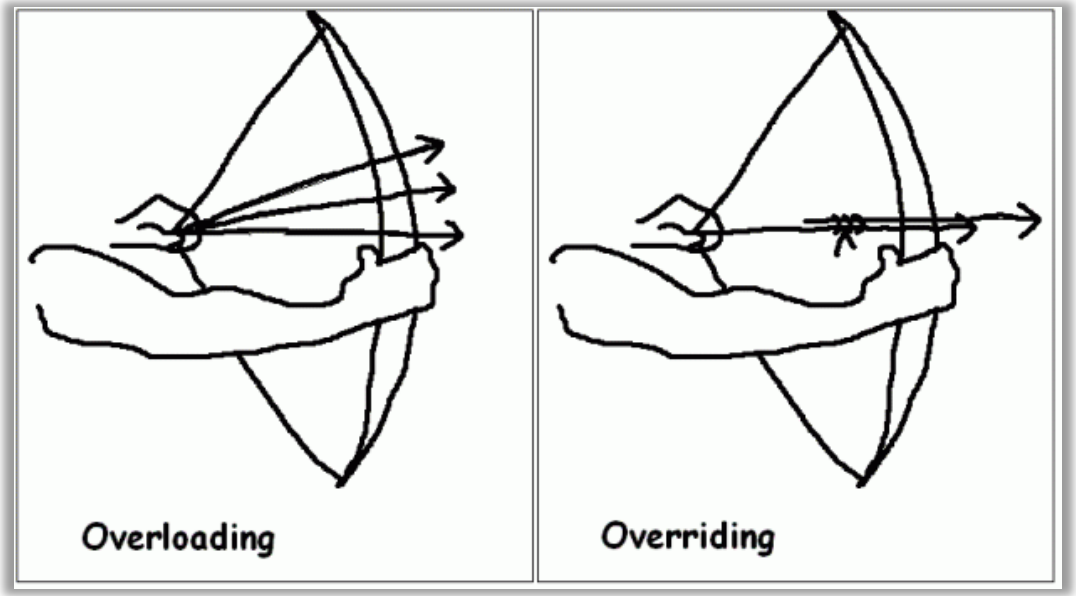
```
class Vehicle:
        def start(self):
            print("Starting engine")
        def stop(self):
            print("Stopping engine")
class TwoWheeler(Vehicle):
        def say(self):
            super().start()
            print("I have two wheels")
            super().stop()
Harley=TwoWheeler()
Harley.say()
```

IntelliPaat

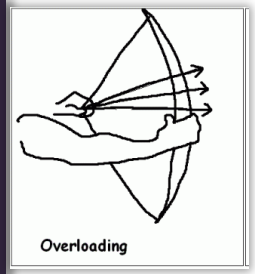
Inheritance in Python

Overriding vs Overloading

"Developers sometimes get confused between them"



Inheritance in Python



Overloading a Function

"Same function with different parameters"

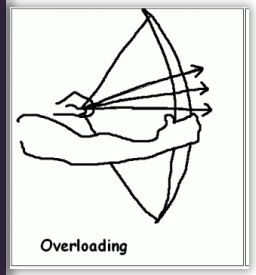
Why Overload a Function

```
def add(a,b):
      return a+b
def add(a,b,c):
      return a+b+c
add(2,3)
```

```
TypeError: add() missing 1
required positional argument: 'c'
```

IntelliPaat

Inheritance in Python



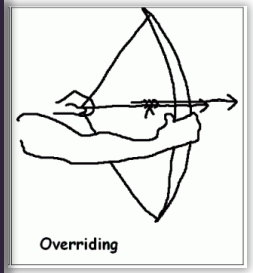
Overloading a Function

"Used to call a method from the parent class"

How to Overload a Function

```
def add(instanceOf,*args):
      if instanceOf=='int':
            result=0
      if instanceOf=='str':
            result=''
      for i in args:
            result+=i
      return result
add('int',3,4,5)
```

Inheritance in Python



Overriding a Function

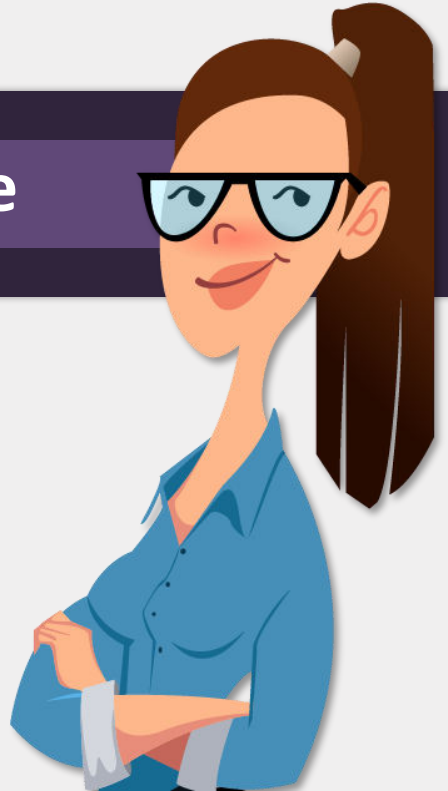
"Subclass may change the functionality of a Python method in the superclass"

Overriding a Function

```
class A:
        def sayhi(self):
                print("I'm in A")
class B(A):
        def sayhi(self):
                print("I'm in B")
bobj=B()
bobj.sayhi()
```

IntelliPaat

Python Certification Course



OOPS: Encapsulation in Python

Things you will learn after this Session

OOPS: Encapsulation in Python

- What is Encapsulation?
- Real life example of Encapsulation
- How to access a private method?
- How to access a private variable?

IntelliPaat

Encapsulation in Python

What is Encapsulation?

Abstraction + Data Hiding

Abstraction is showing essential features and hiding non-essential features to the user.

For example,

While writing a mail you don't know how things are actually happening in the backend

IntelliPaat

Encapsulation in Python

What is Encapsulation?

Abstraction + Data Hiding

Wrapping up of data into a single unit is Encapsulation

For example,

Multiple parts of cars encapsulates itself together to form a single object that is Car

IntelliPaat

Encapsulation in Python

What is Encapsulation?

Convention:

A class variable that should not directly be accessed(private) should be prefixed with an underscore

Encapsulating a Function

```
class Encap(object):
   def __init__(self):
      self.a = 123
      self._b = 123
      self.__c = 123

obj = Encap()
print(obj.a)
print(obj._b)
print(obj.__c)
```

So what's with the underscores and error?

```
AttributeError: 'Encap' object has no attribute '__c'
```

IntelliPaat

Encapsulation in Python

How to access a Private Method?

Private method can be called using redcar._Car__updateSoftware()

Example

```
class Car:

    def __init__(self):
        self.__updateSoftware()

    def drive(self):
        print ('driving')

    def __updateSoftware(self):
        print ('updating software')

redcar = Car()
redcar.drive()
redcar._Car__updateSoftware()
```

IntelliPaat

Encapsulation in Python

How to access a Private Variable?

To change the value of a private variable, a setter method is used

Example

```
    def setMaxSpeed(self,speed):
        self.__maxspeed = speed


redcar = Car()
redcar.drive()
redcar.__maxspeed = 10  # will not change variable
because its private
redcar.setMaxSpeed(320)
redcar.drive()
```

IntelliPaat

OOPS: Polymorphism in Python

Things you will learn after this Session

OOPS: Polymorphism in Python

- What is Polymorphism?
- Real-Life Example of Polymorphism
- Polymorphism in a function

IntelliPaat

Polymorphism in Python

What is Polymorphism?

"Functions with same name, but functioning in different ways"

For Example:

You behave differently in front of elders, and friends.

A single person behaves differently at different time

IntelliPaat

Polymorphism in Python

Polymorphism with a function

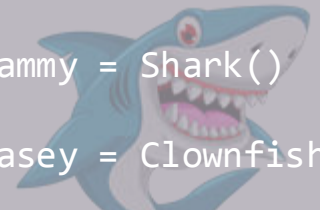
Example

```
def in_the_pacific(fish):
    fish.swim()

sammy = Shark()

casey = Clownfish()

in_the_pacific(sammy)
in_the_pacific(casey)
```



QUIZ

Quiz 1

What is the output of the following?

```
List1 = ['Python', 'Py', 'Pyth', 'Python3']
print(List1[-1][2])
```

A. t

B. 3

C. n

D. Python

Answer 1

What is the output of the following?

```
List1 = ['Python', 'Py', 'Pyth', 'Python3']
print(List1[-1][2])
```

A	t
B	3
C	n
D	Python

Quiz 2

What is the output of the following?

```
x = range(10)
y = sum(x)
print(y)
```

A	10
B	45
C	1
D	An exception is thrown

Answer 2

What is the output of the following?

```
x = range(10)
y = sum(x)
print(y)
```

A	10
B	45
C	1
D	An exception is thrown

Quiz 3

1.What is the output of the following?

```
List1 = ['Python', 'Py', 'Pyth',
'Python3']
List2 = List1*2
List3 = List1[:]
List2[0] = 'Hello'
List3[1] = 'World'
sum = 0
for ls in (List1, List2, List3):
    if ls[0] == 'Hello':
        sum += 1
    if ls[1] == 'World':
        sum += 2
print(sum, List2)
```

A	a)3 ['Hello', 'Py', 'Pyth', 'Python3', 'Python', 'Py', 'Pyth', 'Python3']
B	a)2 ['Hello', 'Py', 'Pyth', 'Python3']
C	a)3 ['Hello', 'Py', 'Pyth', 'Python3']
D	a)2 ['Hello', 'Py', 'Pyth', 'Python3', 'Python', 'Py', 'Pyth', 'Python3']

Answer 3

1.What is the output of the following?

A	a)3 ['Hello', 'Py', 'Pyth', 'Python3', 'Python', 'Py', 'Pyth', 'Python3']
B	a)2 ['Hello', 'Py', 'Pyth', 'Python3']
C	a)3 ['Hello', 'Py', 'Pyth', 'Python3']
D	a)2 ['Hello', 'Py', 'Pyth', 'Python3', 'Python', 'Py', 'Pyth', 'Python3']

```
List1 = ['Python', 'Py', 'Pyth',
'Python3']
List2 = List1*2
List3 = List1[:]
List2[0] = 'Hello'
List3[1] = 'World'
sum = 0
for ls in (List1, List2, List3):
    if ls[0] == 'Hello':
        sum += 1
    if ls[1] == 'World':
        sum += 2
print(sum, List2)
```

Quiz 4

1.We have two lists

a=['a','b']

b=['r','n']

A	a)a.extend(b)
B	a)a.append(b)
C	a)a.insert(1,b)
D	a)None

We want to achieve the following output
['a', 'b', ['r', 'n']]

Which of the following should be used?

Quiz 4

1.We have two lists

a=['a','b']

b=['r','n']

A	a)a.extend(b)
B	a)a.append(b)
C	a)a.insert(1,b)
D	a)None

We want to achieve the following output
['a', 'b', ['r', 'n']]

Which of the following should be used?

Quiz 5

1.What does the following code do?

```
def a(b, c, d): pass
```

A	a)defines a list and initializes it
B	a)defines a function, which does nothing - correct
C	a)defines a function, which passes its parameters through
D	a)defines an empty class

Answer 5

1.What does the following code do?

```
def a(b, c, d): pass
```

A	a)defines a list and initializes it
B	a)defines a function, which does nothing - correct
C	a)defines a function, which passes its parameters through
D	a)defines an empty class

Quiz 6

What is the output?

A. Yes

B. No

C. fails to compiler

```
x = True
y = False
z = False
if x or y and z:
   print("yes")
else:
   print("no")
```

Answer 6

What is the output?

A	Yes
B	No
C	fails to compiler

```
x = True
y = False
z = False
if x or y and z:
   print("yes")
else:
   print("no")
```

IntelliPaat

www.intellipaat.com

CALL US NOW

India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

sales@intellipaat.com