**OOPS CONCEPTS - INHERITANCE**

**INHERITANCE**

Inheritance is the process of defining a new object with the help of an existing object. The 2 key functionality principles are :

1. Accessing Existing Objects Functionality
2. Updating Existing Objects Functionality

Two other terms we usually hear when we talk about inheritance are :

1. Reusability

A diagram of a vehicle

Description automatically generated

1. IS-A Relation

A diagram of a vehicle

Description automatically generated

**Code:**

class **Volvo**:

def **\_\_init\_\_**(*self*, make, model, year):

*self*.carMake=make

*self*.carModel=model

*self*.carYear=year

#INHERITING THE PARENT CLASS IN CHILD CLASS

class **XCFuel**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, seater, fuelType, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carSeater=seater

*self*.carFuelType=fuelType

#CREATE ANOTHER CHILD CLASS AND PROCESS AS ABOVE

class **XCElectric**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, electric, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carElectric=electric

print(*"VOLVO - FUEL TYPE VEHICLE DETAILS"*)

print(*"=================================="*)

xcfuel1 = XCFuel(5,*"Petrol"*,*"Volvo"*,*"XC40"*,2023)

#xcfuel2 = XCFuel(5,"Disel","Volvo","XC40",2023)

#xcfuel3 = XCFuel(7,"Petrol","Volvo","XC60",2024)

#xcfuel4 = XCFuel(7,"Petrol","Volvo","XC90",2024)

print(*"Car Brand : "*, xcfuel1.carMake)

print(*"Car Model : "*, xcfuel1.carModel)

print(*"Car Year : "*, xcfuel1.carYear)

print(*"Car Seater : "*, xcfuel1.carSeater)

print(*"Car Fuel Type : "*, xcfuel1.carFuelType)

print(*"VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS"*)

print(*"=========================================="*)

xelec1 = XCElectric(*"Recharge"*,*"Volvo"*,*"XCE-40"*,2024)

#xelec2 = XCElectric("Recharge","Volvo","XCE-60",2024)

#xelec3 = XCElectric("Recharge","Volvo","XCE-90",2024)

print(*"Car Brand : "*, xelec1.carMake)

print(*"Car Model : "*, xelec1.carModel)

print(*"Car Year : "*, xelec1.carYear)

print(*"Car Fuel Type : "*, xelec1.carElectric)

**Output**

VOLVO - FUEL TYPE VEHICLE DETAILS

==================================

Car Brand : Volvo

Car Model : XC40

Car Year : 2023

Car Seater : 5

Car Fuel Type : Petrol

VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS

==========================================

Car Brand : Volvo

Car Model : XCE-40

Car Year : 2024

Car Fuel Type : Recharge

**INHERITANCE FUNCTIONALITY**

**Code:**

class **Volvo**:

def **\_\_init\_\_**(*self*, make, model, year):

*self*.carMake=make

*self*.carModel=model

*self*.carYear=year

def **carShowroom**(*self*):

print(*"Showrooms are .... 1. JoeDuffy, 2. Volvo"*)

#INHERITING THE PARENT CLASS IN CHILD CLASS

class **XCFuel**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, seater, fuelType, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carSeater=seater

*self*.carFuelType=fuelType

def **carDescription**(*self*):

print(*"Car models available only in Petrol and Disel"*)

#CREATE ANOTHER CHILD CLASS AND PROCESS AS ABOVE

class **XCElectric**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, electric, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carElectric=electric

def **carDescription**(*self*):

print(*"Car models available only in Electric"*)

print(*"VOLVO - FUEL TYPE VEHICLE DETAILS"*)

print(*"=================================="*)

xcfuel1 = XCFuel(5,*"Petrol"*,*"Volvo"*,*"XC40"*,2023)

#xcfuel2 = XCFuel(5,"Disel","Volvo","XC40",2023)

#xcfuel3 = XCFuel(7,"Petrol","Volvo","XC60",2024)

#xcfuel4 = XCFuel(7,"Petrol","Volvo","XC90",2024)

print(*"Car Brand : "*, xcfuel1.carMake)

print(*"Car Model : "*, xcfuel1.carModel)

print(*"Car Year : "*, xcfuel1.carYear)

print(*"Car Seater : "*, xcfuel1.carSeater)

print(*"Car Fuel Type : "*, xcfuel1.carFuelType)

xcfuel1.carShowroom()

xcfuel1.carDescription()

print(*"VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS"*)

print(*"=========================================="*)

xelec1 = XCElectric(*"Recharge"*,*"Volvo"*,*"XCE-90"*,2024)

#xelec2 = XCElectric("Recharge","Volvo","XCE-60",2024)

#xelec3 = XCElectric("Recharge","Volvo","XCE-40",2024)

print(*"Car Brand : "*, xelec1.carMake)

print(*"Car Model : "*, xelec1.carModel)

print(*"Car Year : "*, xelec1.carYear)

print(*"Car Fuel Type : "*, xelec1.carElectric)

xelec1.carShowroom()

xelec1.carDescription()

**Output:**

VOLVO - FUEL TYPE VEHICLE DETAILS

==================================

Car Brand : Volvo

Car Model : XC40

Car Year : 2023

Car Seater : 5

Car Fuel Type : Petrol

Showrooms are .... 1. JoeDuffy, 2. Volvo

Car models available only in Petrol and Disel

VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS

==========================================

Car Brand : Volvo

Car Model : XCE-90

Car Year : 2024

Car Fuel Type : Recharge

Showrooms are .... 1. JoeDuffy, 2. Volvo

Car models available only in Electric

**OVERRIDING FUNCTIONALITY**

**Code:**

class **Volvo**:

def **\_\_init\_\_**(*self*, make, model, year):

*self*.carMake=make

*self*.carModel=model

*self*.carYear=year

def **carShowroom**(*self*):

print(*"Showrooms are .... 1. JoeDuffy, 2. Volvo"*)

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Key"*)

#INHERITING THE PARENT CLASS IN CHILD CLASS

class **XCFuel**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, seater, fuelType, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carSeater=seater

*self*.carFuelType=fuelType

def **carDescription**(*self*):

print(*"Car models available only in Petrol and Disel"*)

#OVERRIDING THE PARENT METHOD IN CHILD CLASS

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Start Stop Button"*)

#CREATE ANOTHER CHILD CLASS AND PROCESS AS ABOVE

class **XCElectric**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, electric, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

Volvo.\_\_init\_\_(*self*, make, model, year)

#CHILD CLASS VALUES

*self*.carElectric=electric

def **carDescription**(*self*):

print(*"Car models available only in Electric"*)

#OVERRIDING THE PARENT METHOD IN CHILD CLASS

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Phone"*)

print(*"VOLVO - FUEL TYPE VEHICLE DETAILS"*)

print(*"=================================="*)

xcfuel1 = XCFuel(5,*"Petrol"*,*"Volvo"*,*"XC40"*,2023)

#xcfuel2 = XCFuel(5,"Disel","Volvo","XC40",2023)

#xcfuel3 = XCFuel(7,"Petrol","Volvo","XC60",2024)

#xcfuel4 = XCFuel(7,"Petrol","Volvo","XC90",2024)

print(*"Car Brand : "*, xcfuel1.carMake)

print(*"Car Model : "*, xcfuel1.carModel)

print(*"Car Year : "*, xcfuel1.carYear)

print(*"Car Seater : "*, xcfuel1.carSeater)

print(*"Car Fuel Type : "*, xcfuel1.carFuelType)

print(*"Calling Class from Parent Class"*)

print(*"-------------------------------"*)

xcfuel1.carShowroom()

print(*"Calling Class from Child Class"*)

print(*"-------------------------------"*)

xcfuel1.carDescription()

print(*"Calling Class from Child Class using overriding"*)

print(*"-----------------------------------------------"*)

xcfuel1.carStartStop()

print(*"VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS"*)

print(*"=========================================="*)

xelec1 = XCElectric(*"Recharge"*,*"Volvo"*,*"XCE-90"*,2024)

#xelec2 = XCElectric("Recharge","Volvo","XCE-60",2024)

#xelec3 = XCElectric("Recharge","Volvo","XCE-40",2024)

print(*"Car Brand : "*, xelec1.carMake)

print(*"Car Model : "*, xelec1.carModel)

print(*"Car Year : "*, xelec1.carYear)

print(*"Car Fuel Type : "*, xelec1.carElectric)

print(*"Calling Class from Parent Class"*)

print(*"-------------------------------"*)

xelec1.carShowroom()

print(*"Calling Class from Child Class"*)

print(*"-------------------------------"*)

xelec1.carDescription()

print(*"Calling Class from Child Class using overriding"*)

print(*"-----------------------------------------------"*)

xelec1.carStartStop()

**Output:**

VOLVO - FUEL TYPE VEHICLE DETAILS

==================================

Car Brand : Volvo

Car Model : XC40

Car Year : 2023

Car Seater : 5

Car Fuel Type : Petrol

Calling Class from Parent Class

-------------------------------

Showrooms are .... 1. JoeDuffy, 2. Volvo

Calling Class from Child Class

-------------------------------

Car models available only in Petrol and Disel

Calling Class from Child Class using overriding

-----------------------------------------------

Car can be Started and Stopped using Start Stop Button

VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS

==========================================

Car Brand : Volvo

Car Model : XCE-90

Car Year : 2024

Car Fuel Type : Recharge

Calling Class from Parent Class

-------------------------------

Showrooms are .... 1. JoeDuffy, 2. Volvo

Calling Class from Child Class

-------------------------------

Car models available only in Electric

Calling Class from Child Class using overriding

-----------------------------------------------

Car can be Started and Stopped using Phone

**SUPER CLASS FUNCTIONALITY**

**Code:**

FROM THE ABOVE CODE IN THE CHILD CLASS

#Volvo.\_\_init\_\_(self, make, model, year)

#SUPER CLASS DELCERATION

super().\_\_init\_\_(make, model, year)

REPLACE BASE CLASS NAME WITH super() AND REMOVE THE self KEYWORD, WHICH WILL GIVE THE SAME OUTPUT

ALSO WE CAN ALSO INVOKE THE PARENT CLASS METHOD FROM CHILD CLASS (BOTH THE SAME FUNCTION NAME)

#OVERRIDING THE PARENT METHOD IN CHILD CLASS

def **carStartStop**(*self*):

#USING SUPER CLASS TO CALL THE PARENT CLASS carStartStop METHOD

super().carStartStop()

# WILL OVERRIDE THE PARENT CLASS carStartStop METHOD BY THIS CLASS

print(*"Car can be Started and Stopped using Start Stop Button"*)

**Output**

Car can be Started and Stopped using Key

Car can be Started and Stopped using Start Stop Button

**Full Code :**

class **Volvo**:

def **\_\_init\_\_**(*self*, make, model, year):

*self*.carMake=make

*self*.carModel=model

*self*.carYear=year

def **carShowroom**(*self*):

print(*"Showrooms are .... 1. JoeDuffy, 2. Volvo"*)

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Key"*)

#INHERITING THE PARENT CLASS IN CHILD CLASS

class **XCFuel**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, seater, fuelType, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

#Volvo.\_\_init\_\_(self, make, model, year)

#SUPER CLASS DELCERATION

super().\_\_init\_\_(make, model, year)

#CHILD CLASS VALUES

*self*.carSeater=seater

*self*.carFuelType=fuelType

def **carDescription**(*self*):

print(*"Car models available only in Petrol and Disel"*)

#OVERRIDING THE PARENT METHOD IN CHILD CLASS

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Start Stop Button"*)

#CREATE ANOTHER CHILD CLASS AND PROCESS AS ABOVE

class **XCElectric**(Volvo):

# ALSO THE 3 FILEDS ON PARENT CLASS

def **\_\_init\_\_**(*self*, electric, make, model, year):

# FIRST LINE OF CHILD CLASS CONSTRUCTOR SHOULD BE TO INVOKE THE PARENT CLASS CONSTRUCTOR

#Volvo.\_\_init\_\_(self, make, model, year)

#SUPER CLASS DELCERATION

super().\_\_init\_\_(make, model, year)

#CHILD CLASS VALUES

*self*.carElectric=electric

def **carDescription**(*self*):

print(*"Car models available only in Electric"*)

#OVERRIDING THE PARENT METHOD IN CHILD CLASS

def **carStartStop**(*self*):

print(*"Car can be Started and Stopped using Phone"*)

print(*"VOLVO - FUEL TYPE VEHICLE DETAILS"*)

print(*"=================================="*)

xcfuel1 = XCFuel(5,*"Petrol"*,*"Volvo"*,*"XC40"*,2023)

#xcfuel2 = XCFuel(5,"Disel","Volvo","XC40",2023)

#xcfuel3 = XCFuel(7,"Petrol","Volvo","XC60",2024)

#xcfuel4 = XCFuel(7,"Petrol","Volvo","XC90",2024)

print(*"Car Brand : "*, xcfuel1.carMake)

print(*"Car Model : "*, xcfuel1.carModel)

print(*"Car Year : "*, xcfuel1.carYear)

print(*"Car Seater : "*, xcfuel1.carSeater)

print(*"Car Fuel Type : "*, xcfuel1.carFuelType)

print(*"Calling Class from Parent Class"*)

print(*"-------------------------------"*)

xcfuel1.carShowroom()

print(*"Calling Class from Child Class"*)

print(*"-------------------------------"*)

xcfuel1.carDescription()

print(*"Calling Class from Child Class using overriding"*)

print(*"-----------------------------------------------"*)

xcfuel1.carStartStop()

print(*"VOLVO - FUEL TYPE ELECTRIC VEHICLE DETAILS"*)

print(*"=========================================="*)

xelec1 = XCElectric(*"Recharge"*,*"Volvo"*,*"XCE-90"*,2024)

#xelec2 = XCElectric("Recharge","Volvo","XCE-60",2024)

#xelec3 = XCElectric("Recharge","Volvo","XCE-40",2024)

print(*"Car Brand : "*, xelec1.carMake)

print(*"Car Model : "*, xelec1.carModel)

print(*"Car Year : "*, xelec1.carYear)

print(*"Car Fuel Type : "*, xelec1.carElectric)

print(*"Calling Class from Parent Class"*)

print(*"-------------------------------"*)

xelec1.carShowroom()

print(*"Calling Class from Child Class"*)

print(*"-------------------------------"*)

xelec1.carDescription()

print(*"Calling Class from Child Class using overriding"*)

print(*"-----------------------------------------------"*)

xelec1.carStartStop()

**QUIZ**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated