class Rectangle:

* def \_\_init\_\_(self, length , width):

self.length = length

self.width = width

* def Perimeter(self):

return 2\*(self.length + self.width)

def Area(self):

return self.length\*self.width

* def display(self):

print("The length of rectangle is: ", self.length)

print("The width of rectangle is: ", self.width)

print("The perimeter of rectangle is: ", self.Perimeter())

print("The area of rectangle is: ", self.Area())

class Parallelepipede(Rectangle):

def \_\_init\_\_(self, length, width , height):

Rectangle.\_\_init\_\_(self, length, width)

self.height = height

* def volume(self):

return self.length\*self.width\*self.height

myRectangle = Rectangle(7 , 5)

myRectangle.display()

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

myParallelepipede = Parallelepipede(7 , 5 , 2)

print("the volume of myParallelepipede is: " , myParallelepipede.volume())

class Person:

* def \_\_init\_\_(self, name, age):

self.name = name

self.age = age

* def display(self):

print("Person name : ", self.name)

print("Person age = ", self.age)

* class Student(Person):

def \_\_init\_\_(self, name , age , section):

Person.\_\_init\_\_(self,name, age)

self.section = section

* def displayStudent(self):

print("Student name : ", self.name)

print("Student age = ", self.age)

print("Student section = ", self.section)

* P = Person("Tomas Wild", 37)

P.display()

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

S = Student("Albert", 23 , "Mathematics")

S.displayStudent()

class BankAccount:

* def \_\_init\_\_(self,accountNumber, name, balance):

self.accountNumber = accountNumber

self.name = name

self.balance = balance

* def Deposit(self , d ):

self.balance = self.balance + d

* def Withdrawal(self , w):

if(self.balance < w):

print("impossible operation! Insufficient balance !")

else:

self.balance = self.balance – w

* def bankFees(self):

self.balance = (95/100)\*self.balance

* def display(self):

print("Account Number : " , self.accountNumber)

print("Account Name : " , self.name)

print("Account Balance : " , self.balance , " $")

* newAccount = BankAccount(2178514584, "Albert" , 2700)

newAccount.Withdrawal(300)

newAccount.Deposit(200)

* newAccount.display()