```
#1. Write a Rectangle class in Python language, allowing you to build a rectangle with
#Create a Perimeter() method to calculate the perimeter of the rectangle and a Area() m
#Create a method display() that display the length, width, perimeter and area of an obj
#Create a Parallelepipede child class inheriting from the Rectangle class and with a he


class rectangle:
    def __init__(self,length,width):
        self.length=length
        self.width=width
    def perimeter(self):
        self.perimeter=2*(self.length*self.width)
        return self.perimeter
    def area(self):
        self.area=(self.length*self.width)
        return self.area
    def display(self):
        print("Length is:",self.length)
        print("width is:",self.width)
        print("perimeter is:",self.perimeter())
        print("area is:",self.area())
class parallelepipede(rectangle):
    def __init__(self,length,width,height):
        rectangle.__init__(self, length, width)
        self.height=height
    def volume(self):
        self.volume=self.length*self.width*self.height
        return volume(self)


abc=rectangle(3,8)
s=parallelepipede(3,8,5)
s.display()
```

```
Length is: 3
width is: 8
perimeter is: 48
area is: 24
```

```
#2.Create a Python class Person with attributes: name and age of type string.Create a d
#and age of an object created via the Person class.Create a child class Student  which
#also has a section attribute.Create a method displayStudent() that displays the name,
#the Student class.Create a student object via an instantiation on the Student class ar

class Person:
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def display(self):
        print("Person name is:",self.name)
        print("Person age is:",self.age)
class student(Person):
    def __init__(self,name,age,section):
        Person.__init__(self,name,age)
        self.section=section
    def display(self):
        print("Person name is:",self.name)
        print("Person age is:",self.age)
        print("section is:",self.section)
a=Person("Priya Talele",24)
s=student("Priya Talele",24,"PYTHON")
s.display()
```

```
Person name is: Priya Talele
Person age is: 24
section is: PYTHON
Person name is: Priya Talele
Person age is: 24
```

```
#3.Create a Python class called BankAccount which represents a bank account, having as
#name (name of the account owner as string type), balance.Create a constructor with par
#Create a Deposit() method which manages the deposit actions.Create a Withdrawal() meth
#Create an bankFees() method to apply the bank fees with a percentage of 5% of the bala
#display account details.Give the complete code for the  BankAccount class.

class Bank_Account:
    def __init__(self):
        self.balance=0
        print("Hello!!! Welcome to the Deposit & Withdrawal Machine")

    def deposit(self):
        amount=float(input("Enter amount to be Deposited: "))
        self.balance += amount
        print("\n Amount Deposited:",amount)

    def withdraw(self):
        amount = float(input("Enter amount to be Withdrawn: "))
        if self.balance>=amount:
            self.balance-=amount
            print("\n You Withdrew:", amount)
        else:
            print("\n Insufficient balance  ")

    def display(self):
        print("\n Net Available Balance=",self.balance)


s = Bank_Account()
s.deposit()
s.withdraw()
s.display()
```

```
Hello!!! Welcome to the Deposit & Withdrawal Machine
Enter amount to be Deposited: 10000

 Amount Deposited: 10000.0
Enter amount to be Withdrawn: 5000

 You Withdrew: 5000.0

 Net Available Balance= 5000.0
```

```python
#4. Define a Book class with the following attributes: Title, Author (Full name), Price
#Define a constructor used to initialize the attributes of the method with values enter
#display information for the current book.Write a program to testing the Book class.---
#-----------------------------------------------------------------------------------------
#-------------------------------------------------------------------------- Create a cla

class Book:
     def __init__(self , Title , Author , Price):
            self.Title=Title
            self.Author=Author
            self.Price=Price
     def view(self):
            return ("Book Title: " ,self.Title , "Book Author: " , self.Author, "Book P

myBook = Book("few things left unsaid" , "Sudeep nagarkar" , "250 Rs")
print(myBook.view())
```

```
('Book Title: ', 'few things left unsaid', 'Book Author: ', 'Sudeep nagarkar', 'Book
Price: ', '250 Rs')
```

```python
#5.Create a child class Bus that will inherit all of the variables and methods of the V
#create relevant methods and variables.2. Define a property that must have the same val
#Define a class attribute "color” with a default value white. I.e., Every Vehicle shoul

class Vehicle:

    def __init__(self, name, max_speed, mileage):
        self.name = name
        self.max_speed = max_speed
        self.mileage = mileage

class Bus(Vehicle):
    pass
```

```
School_bus = Bus("School Volvo", 160, 20)
print("Vehicle name:",School_bus.name,"Speed:",School_bus.max_speed,"Mileage:",School_b
```

```
Vehicle name: School Volvo Speed: 160 Mileage: 20
```