```python
# 1. Write a Python class Car with a constructor to initialize the attributes brand, model, and year.
class Car:
    def __init__(self,b,m,y):
        self.brand=b
        self.model=m
        self.year=y
c=Car("Toyoto",'Tata','2019')



# 2. Create a class Person with a default constructor that initializes the attribute name to "Unknown".
class Person:
    def __init__(self,name="Unknown"):
        self.name=name
        print(self.name)
# p=Person()
# p=Person("lavanya")
# output:
# Unknown
# lavanya


# 3. Write a class Student that initializes name and roll_number via a parameterized constructor.
class Student:
    def __init__(self,name,roll_number):
        self.name=name
        self.rollno=roll_number
        print(self.name)
# s=Student('lavanya',23)


# 4. Create a class Rectangle that initializes length and breadth and includes a method to calculate the area.
class Rectangle:
    def __init__(self,length,breadth):
        self.length=length
        self.breadth=breadth
    def cal_area(self):
        print(self.length*self.breadth)
# r=Rectangle(10,20)
```

```python
# r.cal_area()


# 5. Write a Python program to implement a BankAccount class with attributes account_number, holder_name, and balance, initialized using a constructor.
class BankAccount:
    def __init__(self,acc_no,name,balance):
        self.account_number=acc_no
        self.holder_name=name
        self.balance=balance
        print(self.account_number,self.holder_name,self.balance)
# b=BankAccount(1006,'lavanya','10L')
# output:1006 lavanya 10L


# 6. Create a Circle class with a constructor to initialize radius and a method to calculate the circumference.
class Circle:
    def __init__(self,radius):
        self.radius=radius
    def cal_cir(self):
        print(2*3.14*self.radius)
# c=Circle(2)
# c.cal_cir()
# output:12.56



# 7. Implement a Book class where the constructor initializes title, author, and price, and display these details.

class Book:
    def __init__(self,t,a,p):
        self.title=t
        self.author=a
        self.price=p
    def Display(self):
        print("Title:",self.title,"Author:",self.author,"Price:",self.price)
# b=Book("The Hobbit",'DR Jarvin',9500)
# b.Display()
# output:Title: The Hobbit Author: DR Jarvin Price: 9500
```

```python
# 8. Write a class Employee that initializes name, designation, and salary through the constructor and prints them using a method.

class Employee:
    def __init__(self,n,d,s):
        self.name=n
        self.designation=d
        self.salary=s
    def Display(self):
        print("Name:",self.name,"Designation:",self.designation,"Salary:",self.salary)
# e=Employee("Lavanya","Ds",'10LPA')
# e.Display()
# output:
# Name: Lavanya Designation: Ds Salary: 10LPA




# 9. Create a Laptop class with attributes brand, model, and price, initialized through a constructor. Add a method to apply a discount.
class Laptop:
    def __init__(self,brand,model,price):
        self.brand=brand
        self.model=model
        self.price=price
    def Discount(self):
        print("Brand:",self.brand,"Model:",self.model,"Price:",self.price-10/100*self.price)
# l=Laptop("Toyoto",'Tata',100)
# l.Discount()
# output:
# Brand: Toyoto Model: Tata Price: 90.0


# 10. Write a class Point with attributes x and y initialized via a constructor, and a method to calculate the distance from the origin.
import math
class Point:
    def __init__(self,x,y):
        self.x=x
        self.y=y
    def cal_Distance_from_origin(self,other):
        # print(math.sqrt(((self.x)**2)+((self.y)**2)))
```

```
        print(math.sqrt(((self.x-other.x)**2)+((other.y-other.y)**2)))
p=Point(1,1)
p.cal_Distance_from_origin(Point(0,0))


# 11. Create a class Person with a constructor to initialize name and age. Add a method
is_adult that returns True if the person is 18 or older.
class Person:
    def __init__(self,name,age):
        self.name=name
        self.age=age
    def is_adult(self):
        if self.age>=18:
            return ("True")
# p=Person('lavanya',24)
# print(p.is_adult())
# output:True


# 12. Write a Vehicle class with a constructor to initialize name and max_speed, and add a
method to display these attributes.
class Vehicle:
    def __init__(self,name,max_speed):
        self.name=name
        self.max_speed=max_speed
    def display(self):
        print("Name:",self.name,"Maximum_Speed",self.max_speed)
v=Vehicle('Bajaj',90)
# v.display()
# output:Name: Bajaj Maximum_Speed 90


# 13. Implement a Cube class with attributes side_length initialized through a constructor and
a method to calculate the volume.
class Cube:
    def __init__(self,side_len):
        self.side_len=side_len
    def cal(self):
        print("Volume",self.side_len**3)
# c=Cube(5)
# c.cal()
# output:Volume 125
```

```python
# 14. Write a Student class that initializes a list of marks via a constructor and provides methods to calculate the total and average marks.
class Student:
    def __init__(self,l):
        self.lst=l
        self.sum=0
    def total(self):
        self.sum=sum(self.lst)
        print('Sum:',sum(self.lst))
    def Avg(self):
        print(self.sum/len(self.lst))
l=[10,20,30,40,50]
# s=Student(l)
# s.total()
# s.Avg()
# output:150
# 30.0


# 15. Create a Company class with attributes name, location, and employee_count. Initialize these attributes using a constructor and display the information.
class Company:
    def __init__(self,name,loc,employee_count):
        self.name=name
        self.employee_count=employee_count
        self.loc=loc
    def Display(self):
        print(self.name,self.loc,self.employee_count)
c=Company('lavanya','Solapur',2)
# c.Display()
# output: lavanya Solapur 2
```