1. Write a python script to create a Profile class with 3 attributes (name, email, age).

```
class Profile:


   def __init__(self):
      self.name = "abc"
      self.email ="abc@mail.com"  #instance variable
      self.age = 22


obj1=Profile()
print(obj1.name)
print(obj1.age)
```

2. Write a python script to update the above Profile class with encapsulation.

```
class Profile:


   def __init__(self):
      self.name = "abc"
      self.email = "abc@mail.com"  #instance variable
      self.age = 22


   def setemail(self, t):#updating Email using encapsulation
      self.email = t
   def getemail(self) :
      return self.email


obj1=Profile()
```

```
print(obj1.email)
obj1.setemail("xyz@mail.com")
print(obj1.getemail())
print()
```

3. Write a python script to update 2nd Question, change email and age to __email and __age.

```
class Profile:

    def __init__(self):
        self.name = "abc"
        self.__email = "abc@mail.com"  #instance variable
        self.__age = 22

    def setdata(self, t ,a):#updating Email and age using encapsulation
        self.__email = t
        self.__age = a
    def getdata(self) :
        return self.__email , self.__age

obj1=Profile()
obj1.setdata("abj@mail.com" ,85)
print(obj1.getdata())
obj1.setdata("ab.cd@mail.com",29)
print(obj1.getdata())
print()
```

4. Write a python script to update 2nd Question, add a class variable (platform) and create a classmethod to access it.

```
class Profile:
    name = "abc"
    email ="abc@mail.com"  #class variable
    age = 22
    @classmethod
    def setdata(cls,n,e,a):
        cls.name = n
        cls.email = e
        cls.age = a
    def getdata(cls):
        print( cls.name ,cls.email,cls.age )
ob1=Profile()
print(ob1.name)
ob1.setdata("xyz",ob1.email,ob1.age)
ob1.getdata()
ob1.setdata("xyz",ob1.email,24)
ob1.getdata()
print()
```

5. Write a python script to create a Calculator class with 2 methods for adding and subtracting 2 values.

```
class Calculator:
    def __init__(self,num1,num2):
            self.num1=num1
            self.num2=num2

    def add(self):
        print(  self.num1 + self.num2)

    def sub(self):
        print( self.num1-self.num2)
ob1=Calculator(eval(input("num1: " )),eval(input("num 2 : ")))
ob1.add()
ob1.sub()
print()
```

6. Write a python script to create a Calculator 2.0 class with 2 methods for multiplication and division of 2 values and inherit it from the Calculator class.

```
class Calculator():
    def __init__(self,num1,num2):
            self.num1=num1
            self.num2=num2

    def add(self):
        print(  self.num1 + self.num2)

    def sub(self):
        print( self.num1-self.num2)
```

```
class Calculator2(Calculator):
    def mul(self):
        print(  self.num1 * self.num2)


    def div(self):
        print( self.num1/self.num2)
ob101=Calculator2(eval(input("num1: " )),eval(input("num 2 : ")))
ob101.mul()
ob101.div()
ob101.add()
ob101.sub()
print()
```

7. Write a python script to create a Phone class with 2 methods to print the features (calling and sms).

```
class Phone:


    def __init__(self):
         self.countryCode =int(input("Add Country Code  : "))
         self.contact_no=int(input("Dial a phone no : ") )


    def call(self):
        print(self.countryCode,"-","%d calling...."%self.contact_no)
    def sms(self):
        print("To : ",self.countryCode,"-","%d"%self.contact_no)
        print(input("Type your text message : "))
        print("-----Your message has been sent-----")


dialer=Phone()
dialer.call()
dialer.sms()
print()
```

8. Write a python script to create a SmartPhone class by inheriting Calculator 2.0 and Phone Class.

```
class Calculator():



    def add(self):
        self.num1=eval(input("num 1:"))
        self.num2=eval(input("num 2:"))
        print( self.num1 + self.num2)



    def sub(self):
        self.num1=eval(input("num 1:"))
        self.num2=eval(input("num 2:"))
        print( self.num1-self.num2)


class Calculator2(Calculator):
    def mul(self):
        self.num1=eval(input("num 1:"))
        self.num2=eval(input("num 2:"))
        print(  self.num1 * self.num2)

    def div(self):
        self.num1=eval(input("num 1:"))
        self.num2=eval(input("num 2:"))
        print( self.num1/self.num2)
class Phone:
    def call(self):
         self.countryCode =int(input("Add Country Code  : "))
         self.contact_no=int(input("Dial a phone no : ") )
         print(self.countryCode,"-","%d calling...."%self.contact_no)

    def sms(self):
        self.countryCode =int(input("Add Country Code  : "))
        self.contact_no=int(input("Dial a phone no : ") )
        print("To : ",self.countryCode,"-","%d"%self.contact_no)
        print(input("Type your text message : "))
        print("-----Your message has been sent-----")

class smartPhone(Phone,Calculator2):

    def navigate(self):
        print("opening map to start the navigation")
ob=smartPhone()
ob.call()
ob.mul()
```

9. Write a python script to create an application like Truecaller where names and numbers are stored. Truecaller class will have 2 methods (1st to fetch the name of a number and 2nd to add a new entry).

```
class truecaller:
  def fetchname (self):
    self.people = {"Mom":"222-2222", "Papa":"555-1212", "Joy":"967-1490", "Roy":"333-3333", "Mr.a":"725-3444", "Miss.b":"555-1222", "Mrs.a":"444-4656"}


    self.name =input("Enter person's name:")


    if self.name in self.people:
      print(self.name + " exsist in contact list")
      self.x=int(input("to Call " + self.name + " enter 0 or 1 for sms :"))
      if self.x == 0 :
         print("calling..."+self.name)
      elif self.x==1:
         print("writer your sms to "+self.name)
      else :
        print ("contact list :")
        for i in self.people:
          print (i)
    else:
      print ("your entered name doesn't exsist in your contact list")
      y=input("do you want to add this name in  your phonebook ? [Yes/No]: ")
      if y == "Yes" or y=="yes":
        ob1.addnew()
      else :
        print("Thanks for using Truecaller")
  def addnew(self):
    self.newentries = { input("enter name :") : input("enter phone number :")}
    self.people.update(self.newentries)
    print("Updated contact list is: ",self.people)
ob1=truecaller()
ob1.fetchname()
```

10. Write a python script to add the new method in SmartPhone class which accepts Truecaller object as a parameter and call the fetch method of Truecaller.

```
class Phone:
   def call(self):
       self.countryCode =int(input("Add Country Code  : "))
       self.contact_no=int(input("Dial a phone no : ") )
       print(self.countryCode,"-","%d calling...."%self.contact_no)

   def sms(self):
      self.countryCode =int(input("Add Country Code  : "))
      self.contact_no=int(input("Dial a phone no : ") )
      print("To : ",self.countryCode,"-","%d"%self.contact_no)
      print(input("Type your text message : "))
      print("-----Your message has been sent-----")
class truecaller:
   def fetchname (self):
     self.people = {"Mom":"222-2222", "Papa":"555-1212", "Joy":"967-1490", "Roy":"333-3333", "Mr.a":"725-3444", "Miss.b":"555-1222", "Mrs.a":"444-4656"}


     self.name =input("Enter person's name:")


     if self.name in self.people:
       print(self.name + " exsist in contact list")
       self.x=int(input("to Call " + self.name + " enter 0 or 1 for sms :"))
       if self.x == 0 :
         print("calling..."+self.name)
       elif self.x==1:
         print("writer your sms to "+self.name)
       else :
         print ("contact list :")
         for i in self.people:
           print (i)
     else:
       print ("your entered name doesn't exsist in your contact list")
       y=input("do you want to add this name in  your phonebook ? [Yes/No]: ")
       if y == "Yes" or y=="yes":
         ob3.addnew()
       else :
         print("Thanks for using Truecaller")
   def addnew(self):
     self.newentries = { input("enter name :") : input("enter phone number :")}
```

```
   self.people.update(self.newentries)

   print("Updated contact list is: ",self.people)


class smartPhone(Phone,truecaller):


  def navigate(self):

    print("opening map to start the navigation")

ob3=smartPhone()

ob3.call()

ob3.sms()

ob3.fetchname()
```