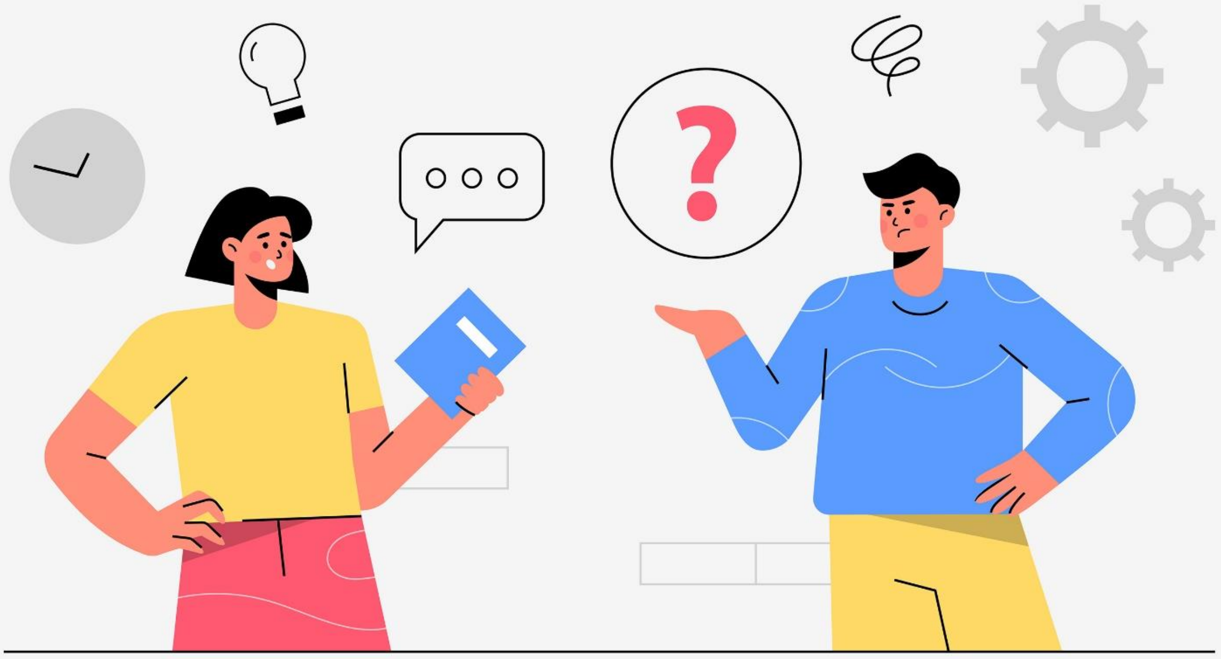
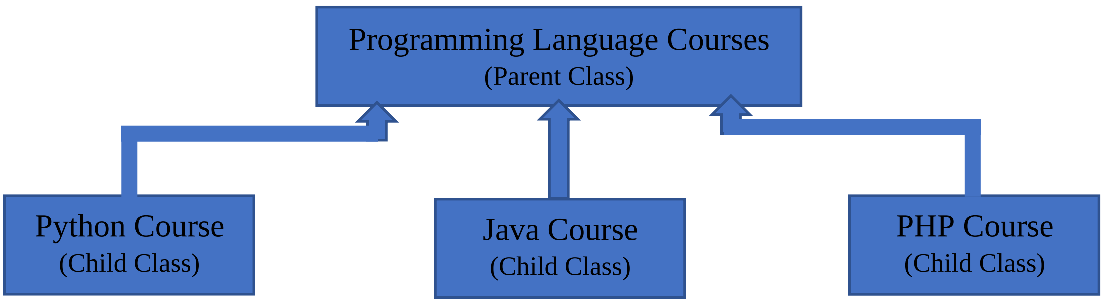
پاسخ سوال برنامه‌نویسی

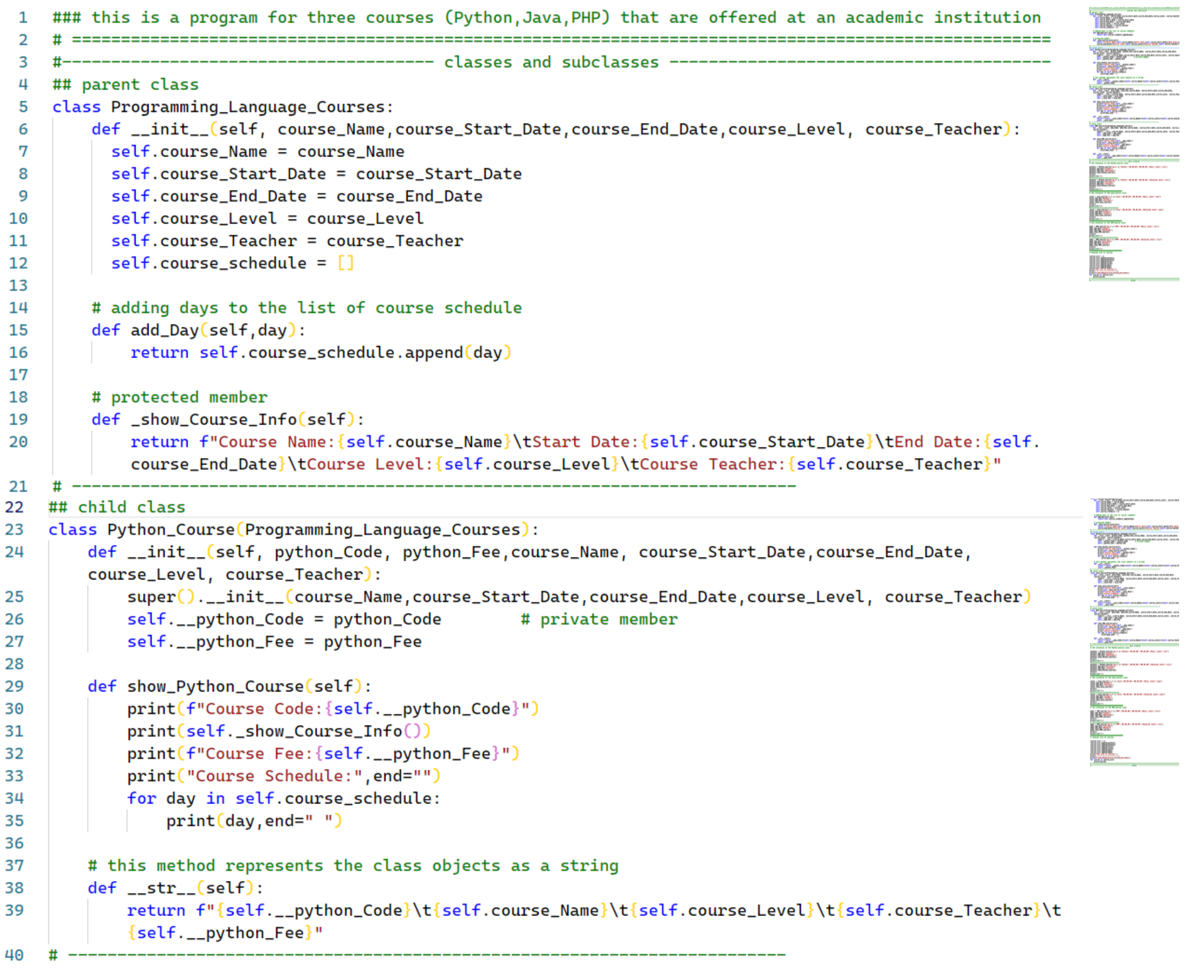
دوره استادی پایتون درسمن

پایتون پیشرفته

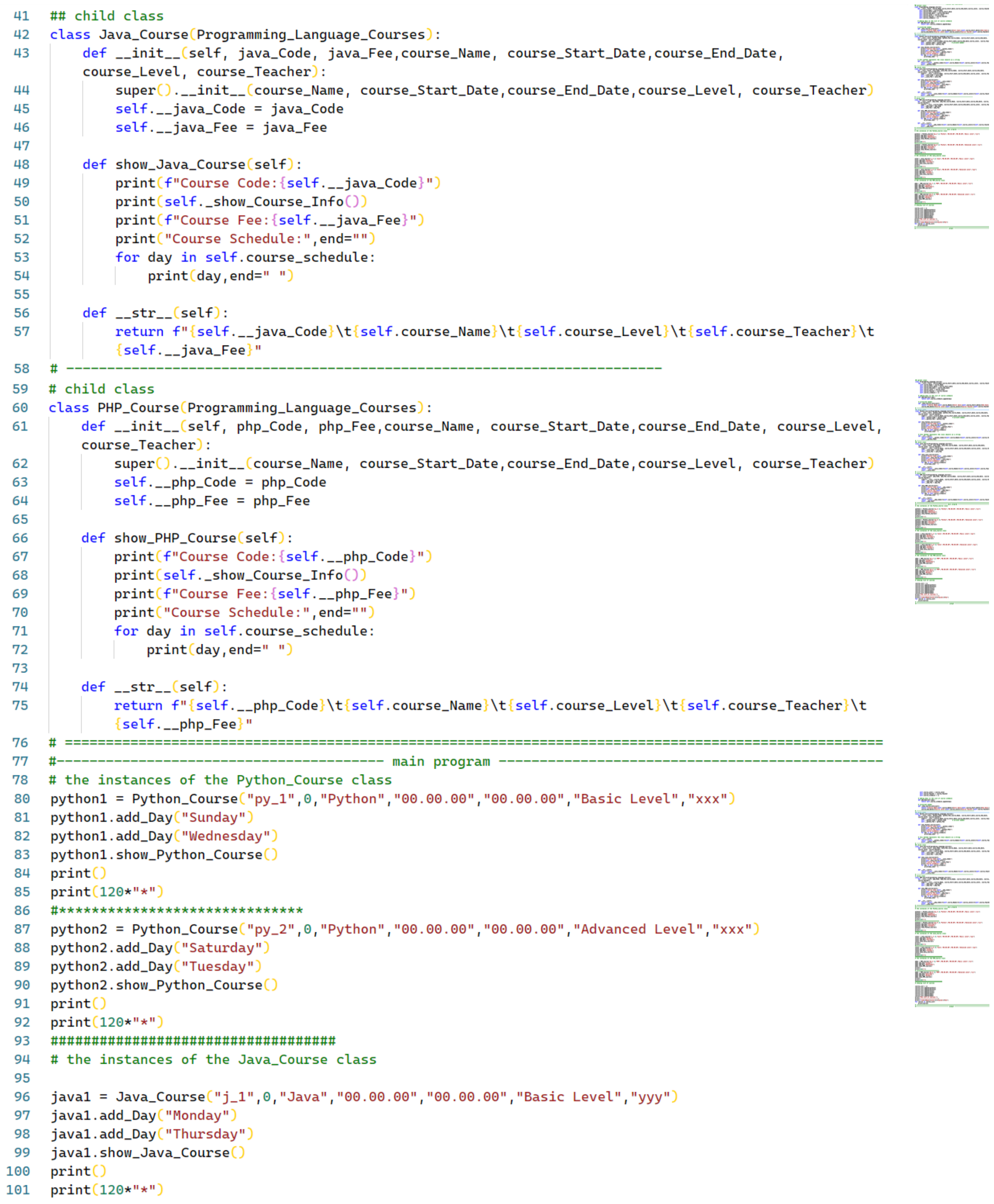


پاسخ تمرین شماره ۱:





```python
### this is a program for three courses (Python,Java,PHP) that are offered at an academic institution
# ======================================================================================================
#-------------------------------------- classes and subclasses ----------------------------------------
## parent class
class Programming_Language_Courses:
    def __init__(self, course_Name,course_Start_Date,course_End_Date,course_Level, course_Teacher):
      self.course_Name = course_Name
      self.course_Start_Date = course_Start_Date
      self.course_End_Date = course_End_Date
      self.course_Level = course_Level
      self.course_Teacher = course_Teacher
      self.course_schedule = []

    # adding days to the list of course schedule
    def add_Day(self,day):
        return self.course_schedule.append(day)

    # protected member
    def _show_Course_Info(self):
        return f"Course Name:{self.course_Name}\tStart Date:{self.course_Start_Date}\tEnd Date:{self.course_End_Date}\tCourse Level:{self.course_Level}\tCourse Teacher:{self.course_Teacher}"
# ---------------------------------------------------------------------------
## child class
class Python_Course(Programming_Language_Courses):
    def __init__(self, python_Code, python_Fee,course_Name, course_Start_Date,course_End_Date, course_Level, course_Teacher):
        super().__init__(course_Name,course_Start_Date,course_End_Date,course_Level, course_Teacher)
        self.__python_Code = python_Code        # private member
        self.__python_Fee = python_Fee

    def show_Python_Course(self):
        print(f"Course Code:{self.__python_Code}")
        print(self._show_Course_Info())
        print(f"Course Fee:{self.__python_Fee}")
        print("Course Schedule:",end="")
        for day in self.course_schedule:
            print(day,end=" ")

    # this method represents the class objects as a string
    def __str__(self):
        return f"{self.__python_Code}\t{self.course_Name}\t{self.course_Level}\t{self.course_Teacher}\t{self.__python_Fee}"
# ---------------------------------------------------------------------------
```

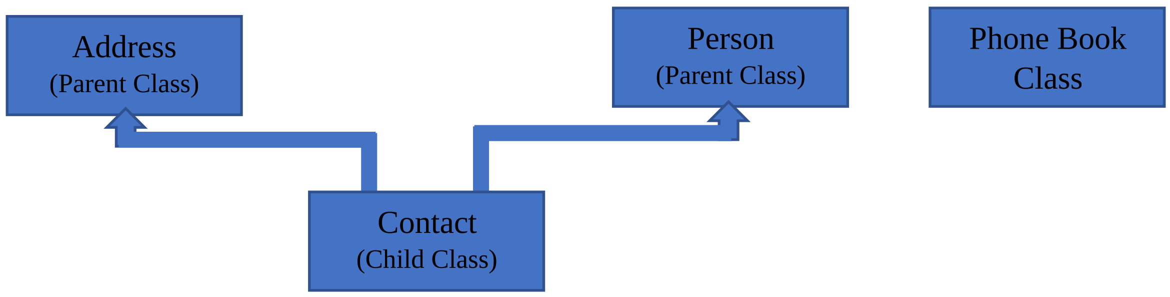


```python
## child class
class Java_Course(Programming_Language_Courses):
    def __init__(self, java_Code, java_Fee,course_Name, course_Start_Date,course_End_Date, course_Level, course_Teacher):
        super().__init__(course_Name, course_Start_Date,course_End_Date,course_Level, course_Teacher)
        self.__java_Code = java_Code
        self.__java_Fee = java_Fee

    def show_Java_Course(self):
        print(f"Course Code:{self.__java_Code}")
        print(self._show_Course_Info())
        print(f"Course Fee:{self.__java_Fee}")
        print("Course Schedule:",end="")
        for day in self.course_schedule:
            print(day,end=" ")

    def __str__(self):
        return f"{self.__java_Code}\t{self.course_Name}\t{self.course_Level}\t{self.course_Teacher}\t{self.__java_Fee}"
# -------------------------------------------------------------------------
# child class
class PHP_Course(Programming_Language_Courses):
    def __init__(self, php_Code, php_Fee,course_Name, course_Start_Date,course_End_Date, course_Level, course_Teacher):
        super().__init__(course_Name, course_Start_Date,course_End_Date,course_Level, course_Teacher)
        self.__php_Code = php_Code
        self.__php_Fee = php_Fee

    def show_PHP_Course(self):
        print(f"Course Code:{self.__php_Code}")
        print(self._show_Course_Info())
        print(f"Course Fee:{self.__php_Fee}")
        print("Course Schedule:",end="")
        for day in self.course_schedule:
            print(day,end=" ")

    def __str__(self):
        return f"{self.__php_Code}\t{self.course_Name}\t{self.course_Level}\t{self.course_Teacher}\t{self.__php_Fee}"
# ====================================================================================================
#---------------------------------------- main program ----------------------------------------------
# the instances of the Python_Course class
python1 = Python_Course("py_1",0,"Python","00.00.00","00.00.00","Basic Level","xxx")
python1.add_Day("Sunday")
python1.add_Day("Wednesday")
python1.show_Python_Course()
print()
print(120*"*")
#******************************
python2 = Python_Course("py_2",0,"Python","00.00.00","00.00.00","Advanced Level","xxx")
python2.add_Day("Saturday")
python2.add_Day("Tuesday")
python2.show_Python_Course()
print()
print(120*"*")
###################################
# the instances of the Java_Course class

java1 = Java_Course("j_1",0,"Java","00.00.00","00.00.00","Basic Level","yyy")
java1.add_Day("Monday")
java1.add_Day("Thursday")
java1.show_Java_Course()
print()
print(120*"*")
```

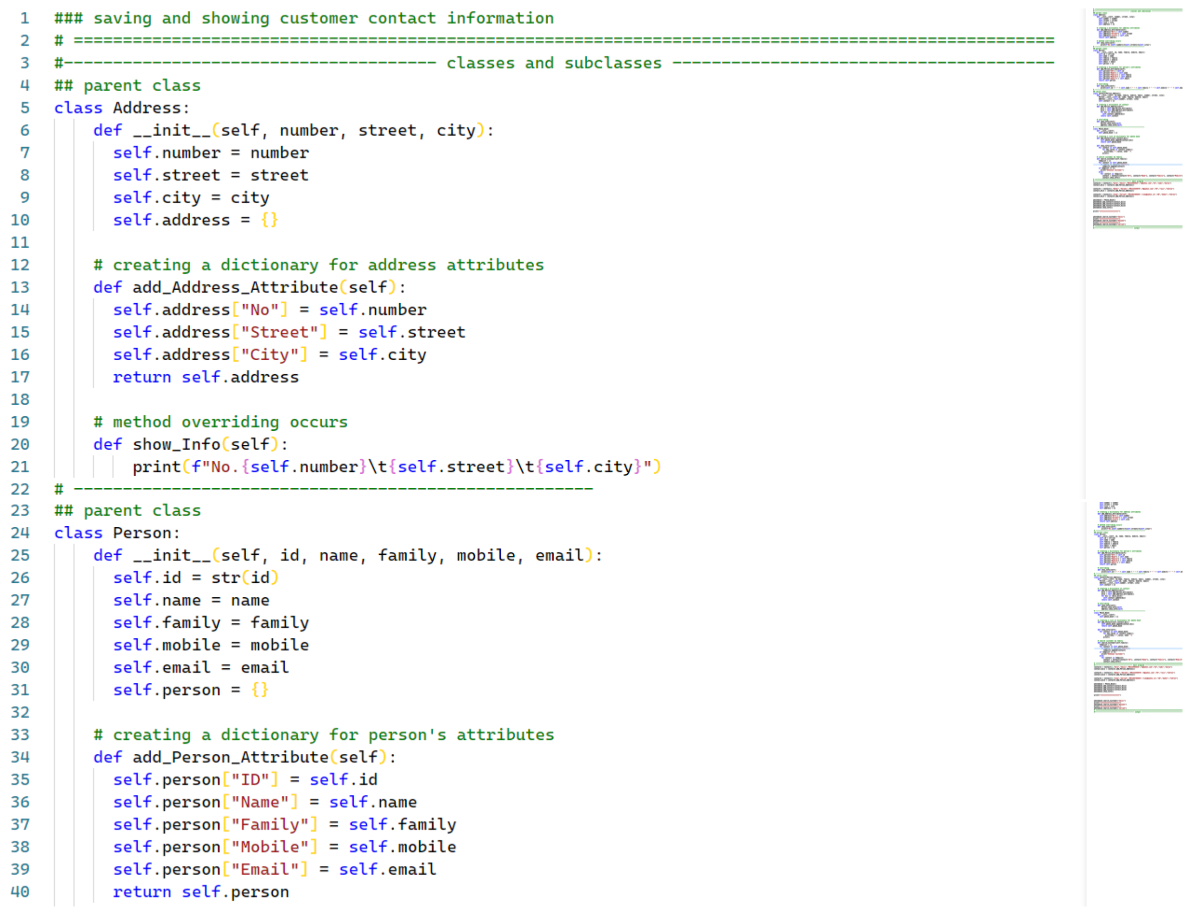
```python
#*******************************
java2 = Java_Course("j_2",0,"Java","00.00.00","00.00.00","Advanced Level","yyy")
java2.add_Day("Sunday")
java2.add_Day("Tuesday")
java2.show_Java_Course()
print()
print(120*"*")
###################################
# the instances of the PHP_Course class

php1 = PHP_Course("ph_1",0,"PHP","00.00.00","00.00.00","Basic Level","zzz")
php1.add_Day("Sunday")
php1.add_Day("Wednesday")
php1.show_PHP_Course()
print()
print(120*"*")
# *****************************
php2 = PHP_Course("ph_2",0,"PHP","00.00.00","00.00.00","Advanced Level","zzz")
php2.add_Day("Saturday")
php2.add_Day("Monday")
php2.show_PHP_Course()
print()
print(120*"*")
###################################
# showing list of courses

course_List = []
course_List.append(python1)
course_List.append(python2)
course_List.append(java1)
course_List.append(java2)
course_List.append(php1)
course_List.append(php2)
print("The list of courses:")
print("----------------------")
print(f"Code\tName\tLevel\t\tTeacher\tFee")
for course in course_List:
    print(course)
# ====================================================================================================
#------------------------------------------ output ---------------------------------------------------
```

```
Course Code:py_1
Course Name:Python      Start Date:00.00.00     End Date:00.00.00       Course Level:Basic Level        Course Teacher:xxx
Course Fee:0
Course Schedule:Sunday Wednesday
************************************************************************************************************************
Course Code:py_2
Course Name:Python      Start Date:00.00.00     End Date:00.00.00       Course Level:Advanced Level     Course Teacher:xxx
Course Fee:0
Course Schedule:Saturday Tuesday
************************************************************************************************************************
Course Code:j_1
Course Name:Java        Start Date:00.00.00     End Date:00.00.00       Course Level:Basic Level        Course Teacher:yyy
Course Fee:0
Course Schedule:Monday Thursday
************************************************************************************************************************
Course Code:j_2
Course Name:Java        Start Date:00.00.00     End Date:00.00.00       Course Level:Advanced Level     Course Teacher:yyy
Course Fee:0
Course Schedule:Sunday Tuesday
************************************************************************************************************************
Course Code:ph_1
Course Name:PHP Start Date:00.00.00     End Date:00.00.00       Course Level:Basic Level        Course Teacher:zzz
Course Fee:0
Course Schedule:Sunday Wednesday
************************************************************************************************************************
Course Code:ph_2
Course Name:PHP Start Date:00.00.00     End Date:00.00.00       Course Level:Advanced Level     Course Teacher:zzz
Course Fee:0
Course Schedule:Saturday Monday
************************************************************************************************************************
The list of courses:
----------------------
Code    Name    Level           Teacher Fee
py_1    Python  Basic Level     xxx     0
py_2    Python  Advanced Level  xxx     0
j_1     Java    Basic Level     yyy     0
j_2     Java    Advanced Level  yyy     0
ph_1    PHP     Basic Level     zzz     0
ph_2    PHP     Advanced Level  zzz     0
```

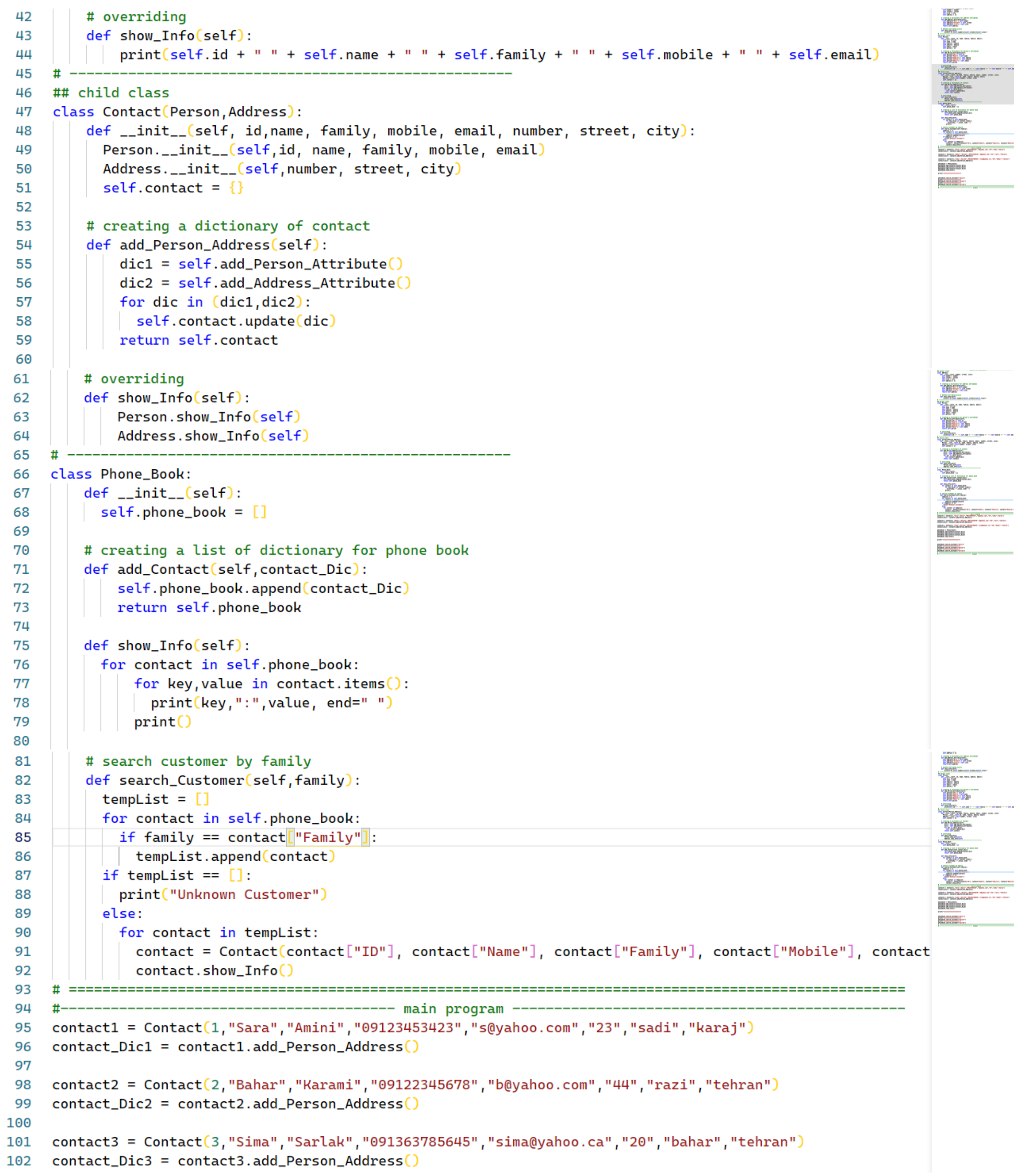
پاسخ تمرین شماره ۲:



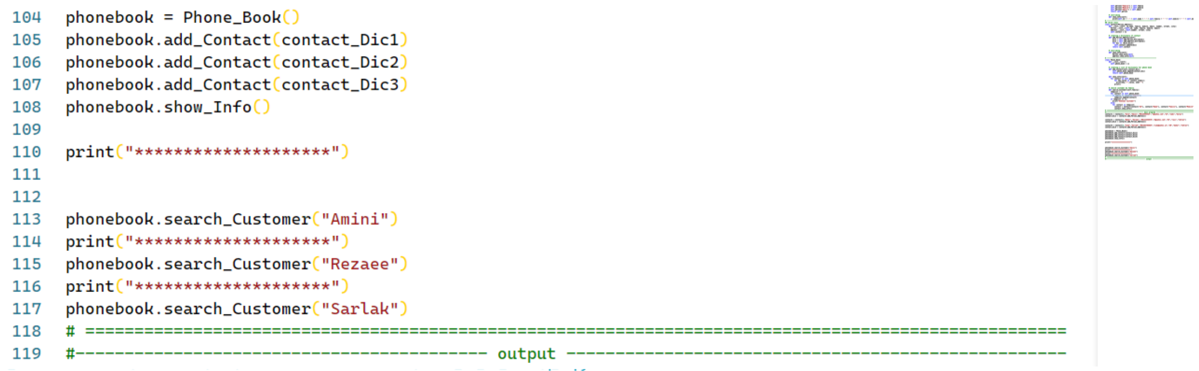
در این راه حل به جای ۱۰ نمونه از کلاس تماس، تنها سه نمونه تعریف شده است، با این هدف که تنها الگویی برای حل این مسأله است.



```python
### saving and showing customer contact information
# ======================================================================================================
#--------------------------------------- classes and subclasses ---------------------------------------
## parent class
class Address:
    def __init__(self, number, street, city):
        self.number = number
        self.street = street
        self.city = city
        self.address = {}

    # creating a dictionary for address attributes
    def add_Address_Attribute(self):
        self.address["No"] = self.number
        self.address["Street"] = self.street
        self.address["City"] = self.city
        return self.address

    # method overriding occurs
    def show_Info(self):
        print(f"No.{self.number}\t{self.street}\t{self.city}")
# -----------------------------------------------------
## parent class
class Person:
    def __init__(self, id, name, family, mobile, email):
        self.id = str(id)
        self.name = name
        self.family = family
        self.mobile = mobile
        self.email = email
        self.person = {}

    # creating a dictionary for person's attributes
    def add_Person_Attribute(self):
        self.person["ID"] = self.id
        self.person["Name"] = self.name
        self.person["Family"] = self.family
        self.person["Mobile"] = self.mobile
        self.person["Email"] = self.email
        return self.person
```



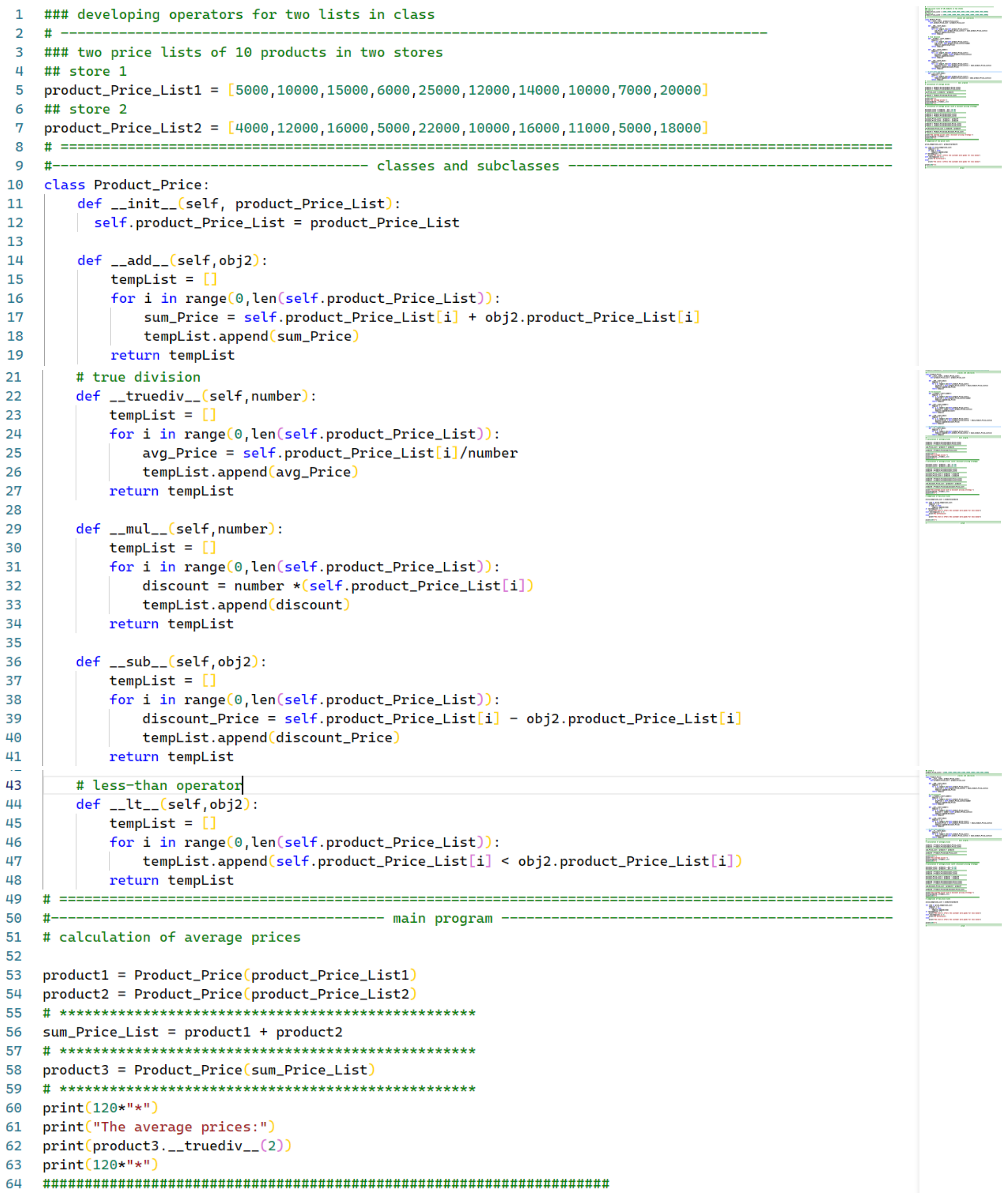
```python
    # overriding
    def show_Info(self):
        print(self.id + " " + self.name + " " + self.family + " " + self.mobile + " " + self.email)
# ---------------------------------------------------
## child class
class Contact(Person,Address):
    def __init__(self, id,name, family, mobile, email, number, street, city):
        Person.__init__(self,id, name, family, mobile, email)
        Address.__init__(self,number, street, city)
        self.contact = {}

    # creating a dictionary of contact
    def add_Person_Address(self):
        dic1 = self.add_Person_Attribute()
        dic2 = self.add_Address_Attribute()
        for dic in (dic1,dic2):
            self.contact.update(dic)
        return self.contact

    # overriding
    def show_Info(self):
        Person.show_Info(self)
        Address.show_Info(self)
# ---------------------------------------------------
class Phone_Book:
    def __init__(self):
        self.phone_book = []

    # creating a list of dictionary for phone book
    def add_Contact(self,contact_Dic):
        self.phone_book.append(contact_Dic)
        return self.phone_book

    def show_Info(self):
        for contact in self.phone_book:
            for key,value in contact.items():
                print(key,":",value, end=" ")
            print()

    # search customer by family
    def search_Customer(self,family):
        tempList = []
        for contact in self.phone_book:
            if family == contact["Family"]:
                tempList.append(contact)
        if tempList == []:
            print("Unknown Customer")
        else:
            for contact in tempList:
                contact = Contact(contact["ID"], contact["Name"], contact["Family"], contact["Mobile"], contact
                contact.show_Info()
# ====================================================================================================
#---------------------------------------- main program ---------------------------------------------
contact1 = Contact(1,"Sara","Amini","09123453423","s@yahoo.com","23","sadi","karaj")
contact_Dic1 = contact1.add_Person_Address()

contact2 = Contact(2,"Bahar","Karami","09122345678","b@yahoo.com","44","razi","tehran")
contact_Dic2 = contact2.add_Person_Address()

contact3 = Contact(3,"Sima","Sarlak","091363785645","sima@yahoo.ca","20","bahar","tehran")
contact_Dic3 = contact3.add_Person_Address()
```



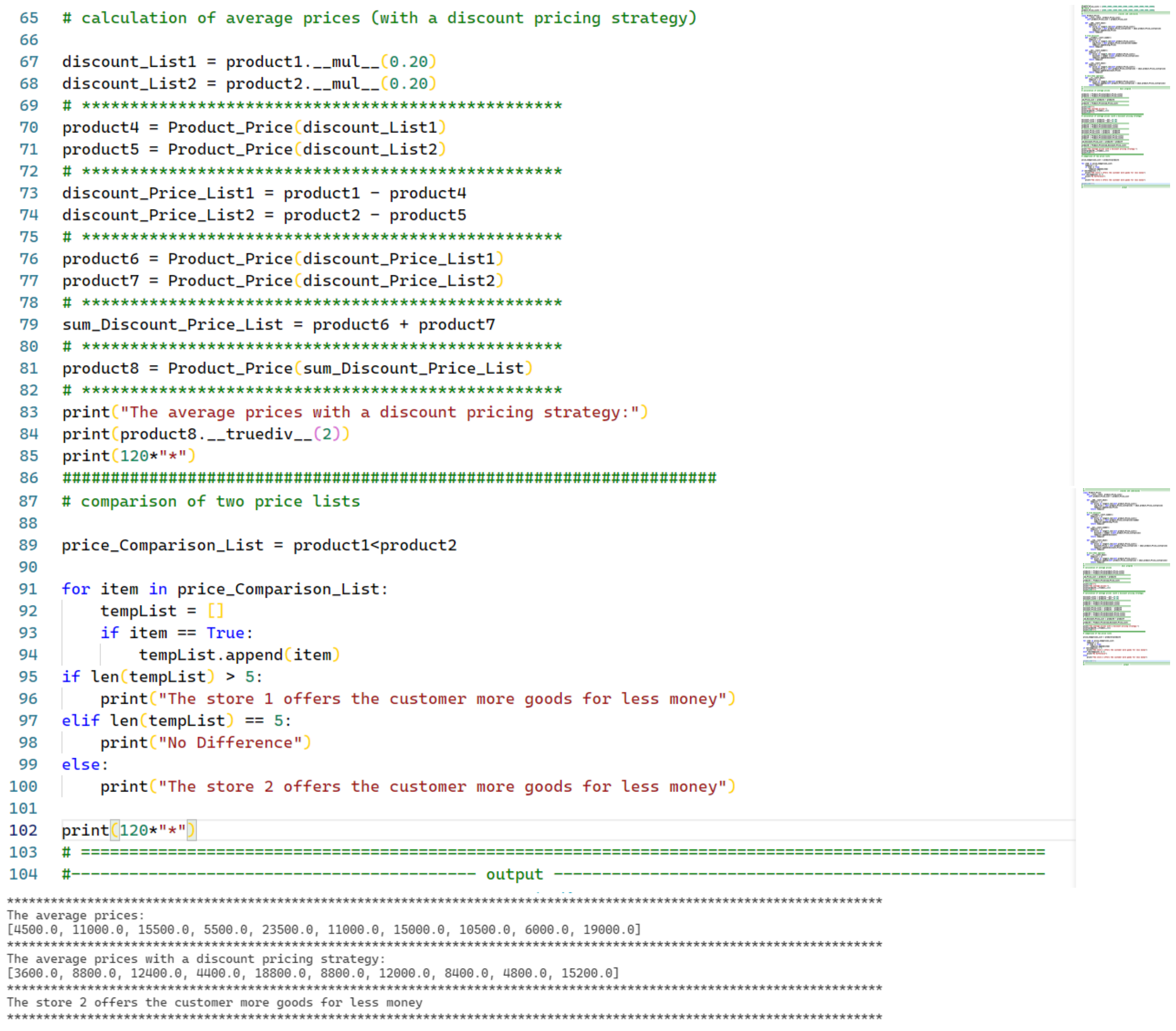
```python
phonebook = Phone_Book()
phonebook.add_Contact(contact_Dic1)
phonebook.add_Contact(contact_Dic2)
phonebook.add_Contact(contact_Dic3)
phonebook.show_Info()

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


phonebook.search_Customer("Amini")
print("********************")
phonebook.search_Customer("Rezaee")
print("********************")
phonebook.search_Customer("Sarlak")
# ==============================================================================================
#----------------------------------------- output -------------------------------------------------
```

```
ID : 1 Name : Sara Family : Amini Mobile : 09123453423 Email : s@yahoo.com No : 23 Street : sadi City : karaj
ID : 2 Name : Bahar Family : Karami Mobile : 09122345678 Email : b@yahoo.com No : 44 Street : razi City : tehran
ID : 3 Name : Sima Family : Sarlak Mobile : 091363785645 Email : sima@yahoo.ca No : 20 Street : bahar City : tehran
********************
1 Sara Amini 09123453423 s@yahoo.com
No.23   sadi    karaj
********************
Unknown Customer
********************
3 Sima Sarlak 091363785645 sima@yahoo.ca
No.20   bahar   tehran
```

پاسخ تمرین شماره ۳:



```python
### developing operators for two lists in class
# -------------------------------------------------------------------------------------
### two price lists of 10 products in two stores
## store 1
product_Price_List1 = [5000,10000,15000,6000,25000,12000,14000,10000,7000,20000]
## store 2
product_Price_List2 = [4000,12000,16000,5000,22000,10000,16000,11000,5000,18000]
# ====================================================================================================
#-------------------------------------- classes and subclasses ---------------------------------------
class Product_Price:
    def __init__(self, product_Price_List):
      self.product_Price_List = product_Price_List

    def __add__(self,obj2):
        tempList = []
        for i in range(0,len(self.product_Price_List)):
            sum_Price = self.product_Price_List[i] + obj2.product_Price_List[i]
            templist.append(sum_Price)
        return tempList
    # true division
    def __truediv__(self,number):
        tempList = []
        for i in range(0,len(self.product_Price_List)):
            avg_Price = self.product_Price_List[i]/number
            tempList.append(avg_Price)
        return tempList

    def __mul__(self,number):
        tempList = []
        for i in range(0,len(self.product_Price_List)):
            discount = number *(self.product_Price_List[i])
            tempList.append(discount)
        return tempList

    def __sub__(self,obj2):
        tempList = []
        for i in range(0,len(self.product_Price_List)):
            discount_Price = self.product_Price_List[i] - obj2.product_Price_List[i]
            tempList.append(discount_Price)
        return tempList
    # less-than operator
    def __lt__(self,obj2):
        tempList = []
        for i in range(0,len(self.product_Price_List)):
            tempList.append(self.product_Price_List[i] < obj2.product_Price_List[i])
        return tempList
# ====================================================================================================
#---------------------------------------- main program -----------------------------------------------
# calculation of average prices

product1 = Product_Price(product_Price_List1)
product2 = Product_Price(product_Price_List2)
# **************************************************
sum_Price_List = product1 + product2
# **************************************************
product3 = Product_Price(sum_Price_List)
# **************************************************
print(120*"*")
print("The average prices:")
print(product3.__truediv__(2))
print(120*"*")
####################################################################
```



```python
# calculation of average prices (with a discount pricing strategy)

discount_List1 = product1.__mul__(0.20)
discount_List2 = product2.__mul__(0.20)
# **************************************************
product4 = Product_Price(discount_List1)
product5 = Product_Price(discount_List2)
# **************************************************
discount_Price_List1 = product1 - product4
discount_Price_List2 = product2 - product5
# **************************************************
product6 = Product_Price(discount_Price_List1)
product7 = Product_Price(discount_Price_List2)
# **************************************************
sum_Discount_Price_List = product6 + product7
# **************************************************
product8 = Product_Price(sum_Discount_Price_List)
# **************************************************
print("The average prices with a discount pricing strategy:")
print(product8.__truediv__(2))
print(120*"*")
####################################################################
# comparison of two price lists

price_Comparison_List = product1<product2

for item in price_Comparison_List:
    tempList = []
    if item == True:
        tempList.append(item)
if len(tempList) > 5:
    print("The store 1 offers the customer more goods for less money")
elif len(tempList) == 5:
    print("No Difference")
else:
    print("The store 2 offers the customer more goods for less money")

print(120*"*")
# =============================================================================================================
#------------------------------------------- output ---------------------------------------------------
```

```
************************************************************************************************************************
The average prices:
[4500.0, 11000.0, 15500.0, 5500.0, 23500.0, 11000.0, 15000.0, 10500.0, 6000.0, 19000.0]
************************************************************************************************************************
The average prices with a discount pricing strategy:
[3600.0, 8800.0, 12400.0, 4400.0, 18800.0, 8800.0, 12000.0, 8400.0, 4800.0, 15200.0]
************************************************************************************************************************
The store 2 offers the customer more goods for less money
************************************************************************************************************************
```