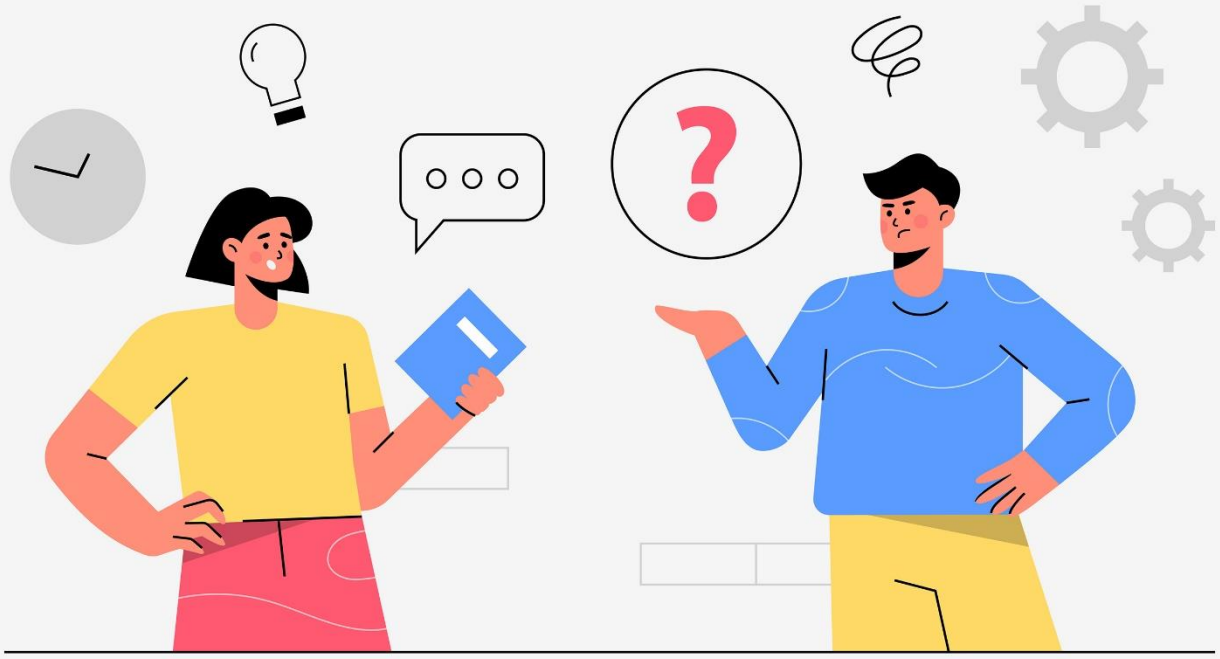
پاسخ سوال برنامه‌نویسی

دوره استادی پایتون درسمن

پایتون پیشرفته



پاسخ تمرین شماره ۱:

در این راه حل به جای ۱۰ نمونه از کلاس دانش‌آموز، تنها ۵ نمونه تعریف شده است، با این هدف که تنها الگویی برای حل این مسأله است.

```python
### class for student in school
# ================================================================================
class Student:
  school_Name = "SABA School"   # class variable

  def __init__(self, student_Code,name, family,level):     # instance method
      self.student_Code = student_Code
      self.name = name
      self.family = family
      self.level = level

  # this function shows list of courses for each level
  @staticmethod
  def show_Courses_By_Level(level):
    if level == 1:
      courses_List1=["C1","C2","C3","C4","C5"]   # عنوان درس‌های پایه اول ابتدایی
      for course in courses_List1:
        print(course,end=" ")
    elif level == 2:
      courses_List2=["C1","C2","C3","C4","C5","C6","C7","C8"]   # عنوان درس‌های پایه دوم ابتدایی
      for course in courses_List2:
        print(course,end=" ")
    elif level == 3:
      courses_List2=["C1","C2","C3","C4","C5","C6","C7","C8","C9"]   # عنوان درس‌های پایه سوم ابتدایی
      for course in courses_List2:
        print(course,end=" ")
    elif level == 4:
      courses_List2=["C1","C2","C3","C4","C5","C6","C7","C8","C9","C10"]   # عنوان درس‌های پایه چهارم ابتدایی
      for course in courses_List2:
        print(course,end=" ")
    elif level == 5:
      courses_List2=["C1","C2","C3","C4","C5","C6","C7","C8","C9"]   # عنوان درس‌های پایه پنجم ابتدایی
      for course in courses_List2:
        print(course,end=" ")
    elif level == 6:
      courses_List2=["C1","C2","C3","C4","C5","C6","C7","C8","C9","C10","C11"]   # عنوان درس‌های پایه ششم ابتدایی
      for course in courses_List2:
        print(course,end=" ")

  # this function changes value of class variable
  @classmethod
  def change_School_Name (Student,new_Name):
    Student.school_Name = new_Name
    return Student.school_Name

  # this function creats a hash code for each object
  def __hash__(self):    # instance method
    return hash(self.student_Code)+ hash(self.name) + hash(self.family) + hash(self.level)

  # this function compares the instances of the class
  def __eq__(self,obj2):       # instance method
    if not isinstance(obj2,Student):
      return False
    return self.student_Code == obj2.student_Code and self.name ==obj2.name and self.family == obj2.famil

  def __str__(self):       # instance method
    return f"Code:{self.student_Code}\tName:{self.name}\tFamily:{self.family}\tLevel:{self.level}"
# ================================================================================================
#---------------------------------------- main program ------------------------------------------
students_Set = set()
for i in range(5):
  code = int(input("Enter Student Code: "))
  name = input("Enter Student Name: ")
  family = input("Enter Student family: ")
  level = int(input("Enter Student Level: "))
  student = Student(code, name, family, level)
  students_Set.add(student)
```

```python
    print("-------------------------")

print("List of Students:\n")
for student in students_Set:
    print(student)
print("-------------------------")
#########################################################
level = input("Enter Level: ")
print("-------------------------")
print(f"List of Courses for Level({level}):\n")
Student.show_Courses_By_Level(3)
print()
print("-------------------------")
#########################################################
print("School Name changed to",Student.change_School_Name ("Narges"),"School")
print("-------------------------")
# ================================================================================================
#---------------------------------------- output ----------------------------------------------
```

```
Enter Student Code: 1
Enter Student Name: Sara
Enter Student family: Salimi
Enter Student Level: 3
-------------------------
Enter Student Code: 2
Enter Student Name: Sima
Enter Student family: Sadr
Enter Student Level: 5
-------------------------
Enter Student Code: 3
Enter Student Name: Bahar
Enter Student family: Karami
Enter Student Level: 6
-------------------------
Enter Student Code: 1
Enter Student Name: Sara
Enter Student family: Salimi
Enter Student Level: 3
-------------------------
Enter Student Code: 4
Enter Student Name: Maryam
Enter Student family: Majd
Enter Student Level: 2
-------------------------
List of Students:

Code:3  Name:Bahar      Family:Karami   Level:6
Code:4  Name:Maryam     Family:Majd     Level:2
Code:2  Name:Sima       Family:Sadr     Level:5
Code:1  Name:Sara       Family:Salimi   Level:3
-------------------------
Enter Level: 5
-------------------------
List of Courses for Level(5):

C1 C2 C3 C4 C5 C6 C7 C8 C9
-------------------------
School Name changed to Narges School
-------------------------
```

پاسخ تمرین شماره ۲:

```python
### class for participants
# ================================================================================
from abc import ABC,abstractmethod
# ================================================================================
#-------------------------------- classes and subclasses ----------------------------
## parent class
class Participants(ABC):
    def __init__(self, name, family, id_Number, major, address):
      self.name = name
      self.family = family
      self.id_Number = id_Number
      self.major = major
      self.address = address

    @abstractmethod
    def calculation_Score(self):
        pass

    def _show_Participant_Information(self):
        return f"Name:{self.name}\tFamily:{self.family}\tID Number:{self.id_Number}\tMajor:{self.major}
        \tAddress:{self.address}"
# ---------------------------------------------------------------------------
## child class
class Participant_Free(Participants):        # شرکت کننده آزاد
    def __init__(self, pf_Code, name, family, id_Number, major, address,test_Score1, test_Score2):
        Participants.__init__(self,name, family, id_Number, major, address)
        self.__pf_Code = pf_Code
        self.__test_Score1 = test_Score1    #نمره آزمون کتبی
        self.__test_Score2 = test_Score2    #نمره مصاحبه


    def calculation_Score(self):
        final_Score = ( self.__test_Score1 + self.__test_Score2)/2
        return final_Score

    @property
    def test_Score1(self):
        return self.__test_Score1

    @test_Score1.setter
    def test_Score1(self,new_Score):
        self.__test_Score1 = new_Score
    @property
    def test_Score2(self):
        return self.__test_Score2

    @test_Score2.setter
    def test_Score2(self,new_Score):
        self.__test_Score2 = new_Score

    def __str__(self):
        return f"Code:{self.__pf_Code}\t{self._show_Participant_Information()}\tScore:{self.
        calculation_Score()}"
# ---------------------------------------------------------------------------
## child class
class Participants_Special(Participants):      # دانش آموخته نخبه

    def __init__(self, ps_Code, name, family, id_Number, major, address, university_Rank,
    grade_Point_Average):
        Participants.__init__(self,name, family, id_Number, major, address)
        self.__ps_Code = ps_Code
        self.__university_Rank = university_Rank
        self.__grade_Point_Average = grade_Point_Average
```

```python
    def calculation_Score(self):
        if self.__university_Rank == 1 and self.__grade_Point_Average >= 18.5:
            return (100+100)/2

        elif self.__university_Rank == 1 and 17.5 <= self.__grade_Point_Average < 18.5:
            return (100+80)/2

        elif self.__university_Rank == 1 and 16 < self.__grade_Point_Average < 17.5:
            return (100+60)/2

        elif self.__university_Rank == 2 and  self.__grade_Point_Average >= 18.5:
            return (80+100)/2

        elif self.__university_Rank == 2 and 17.5 <= self.__grade_Point_Average < 18.5:
            return (80+80)/2

        elif self.__university_Rank == 2 and 16 < self.__grade_Point_Average < 17.5:
            return (80+60)/2

        elif self.__university_Rank == 3 and  self.__grade_Point_Average >= 18.5:
            return (60+100)/2

        elif self.__university_Rank == 3 and 17.5 <= self.__grade_Point_Average < 18.5:
            return (60+80)/2

        elif self.__university_Rank == 3 and 16 < self.__grade_Point_Average < 18.5:
            return (60+60)/2

    @property
    def university_Rank(self):
        return self.__university_Rank

    @university_Rank.setter
    def university_Rank(self,new_Rank):
        self.__university_Rank = new_Rank

    @property
    def grade_Point_Average(self):
        return self.__grade_Point_Average

    @grade_Point_Average.setter
    def grade_Point_Average(self,new_Average):
        self.__grade_Point_Average = new_Average
    def __str__(self):
        return f"Code:{self.__ps_Code}\t{self._show_Participant_Information()}\tScore:{self.
        calculation_Score()}"
# ------------------------------------------------------------------------
## child class
class Employee(Participants):        # کارمند قراردادی
    def __init__(self, e_Code, name, family, id_Number, major, address, performance_Score,
    working_Years):
        Participants.__init__(self,name, family, id_Number, major, address)
        self.__e_Code = e_Code
        self.__performance_Score = performance_Score
        self.__working_Years = working_Years

    def calculation_Score(self):
        if 1< self.__working_Years <= 5:
            final_Score = self.__performance_Score + (self.__performance_Score * 0.1)
            return final_Score
        else:
            final_Score = self.__performance_Score + (self.__performance_Score * 0.2)
            return final_Score
```

```python
    @property
    def performance_Score(self):
        return self.__performance_Score

    @performance_Score.setter
    def performance_Score(self,new_Score):
        self.__performance_Score = new_Score

    @property
    def working_Years(self):
        return self.__working_Years

    @working_Years.setter
    def working_Years(self,new_Number):
        self.__working_Years = new_Number

    def __str__(self):
        return f"Code:{self.__e_Code}\t{self._show_Participant_Information()}\tScore:{self.
        calculation_Score()}"
# ==========================================================================================================
#---------------------------------------- main program ------------------------------------------------------
participants_List = []
accepted_Participants_List = []
##############################################################################################

# the instances of the Participant_Free class

for i in range(3):
    code = input("Enter Code: ")
    name = input("Enter Name: ")
    family = input("Enter Family: ")
    id_Number = input("Enter Id Number: ")
    major = input("Enter Major: ")
    address = input("Enter Address: ")
    score1 = int(input("Enter Test Score1: "))
    score2 = int(input("Enter Test Score2: "))
    pf = Participant_Free(code,name,family,id_Number,major,address,score1,score2)
    participants_List.append(pf)
    if pf.calculation_Score() >= 90:
        accepted_Participants_List.append(pf)
    print("--------------------")
##############################################################################################

# the instances of the Participants_Special class

for i in range(3):
    code = input("Enter Code: ")
    name = input("Enter Name: ")
    family = input("Enter Family: ")
    id_Number = input("Enter Id Number: ")
    major = input("Enter Major: ")
    address = input("Enter Address: ")
    university = int(input("Enter University Rank: "))
    average = float(input("Enter GPA: "))
    ps = Participants_Special(code,name,family,id_Number,major,address,university,average)
    participants_List.append(ps)
    if ps.calculation_Score() >= 90:
        accepted_Participants_List.append(ps)
    print("--------------------")

##############################################################################################

# the instances of the Employee class

for i in range(3):
    code = input("Enter Code: ")
    name = input("Enter Name: ")
    family = input("Enter Family: ")
    id_Number = input("Enter Id Number: ")
    major = input("Enter Major: ")
    address = input("Enter Address: ")
    performance = int(input("Enter Performance Score: "))
    working_years = float(input("Enter Working Years: "))
    e = Employee(code,name,family,id_Number,major,address,performance,working_years)
    participants_List.append(e)
    if e.calculation_Score() >= 90:
        accepted_Participants_List.append(e)
    print("--------------------")


##############################################################################################
```




```
206
207  print("List of Participants:\n")
208  for participant in participants_List:
209      print(participant)
210  print(100*"*")
211  print("List of Accepted Participants:\n")
212  for participant in accepted_Participants_List:
213      print(participant)
214  # ============================================================================================
215  #---------------------------------------- output ----------------------------------------------
```

```
Enter Code: pf-1
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Test Score1: 80
Enter Test Score2: 97
--------------------
Enter Code: pf-2
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Test Score1: 70
Enter Test Score2: 80
--------------------
Enter Code: pf-3
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Test Score1: 85
Enter Test Score2: 90
--------------------
Enter Code: ps-1
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter University Rank: 3
Enter GPA: 18.5
--------------------
Enter Code: ps-2
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter University Rank: 1
Enter GPA: 19
--------------------
Enter Code: ps-3
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter University Rank: 2
Enter GPA: 19
--------------------
Enter Code: ee-1
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Performance Score: 88
Enter Working Years: 5
--------------------
Enter Code: ee-2
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Performance Score: 90
Enter Working Years: 3
--------------------
Enter Code: ee-3
Enter Name: aa
Enter Family: bb
Enter Id Number: 000
Enter Major: cc
Enter Address: dd
Enter Performance Score: 70
Enter Working Years: 6
--------------------
List of Participants:

Code:pf-1       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:88.5
Code:pf-2       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:75.0
Code:pf-3       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:87.5
Code:ps-1       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:80.0
Code:ps-2       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:100.0
Code:ps-3       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:90.0
Code:ee-1       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:96.8
Code:ee-2       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:99.0
Code:ee-3       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:84.0
****************************************************************************************************
List of Accepted Participants:

Code:ps-2       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:100.0
Code:ps-3       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:90.0
Code:ee-1       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:96.8
Code:ee-2       Name:aa Family:bb       ID Number:000   Major:cc        Address:dd      Score:99.0
```