WEEK6: Assignment- Python OOP

Q1.Problem on class and object Task Write a Person class with an instance variable, age, and a constructor that takes an integer, initialAge,as a parameter. The constructor must assign initialAge to age after confirming the argument passed as initialAge is not negative;if a negative argument is passed as initialAge,the constructor should set age to 0 and print Age is not valid,setting age to 0. In addition, you must write the following instance methods: 1.yearPasses() should increase the age instance variable by 1 2. amIOld() should perform the following conditional actions: - If age < 13, print You are young.. - If age > 13 and age < 18, print You are a teenager - Otherwise, print You are old.. Input Format The first line contains an integer,T(the number of test cases), and the T subsequent lines each contain an integer denoting the age of a Person instance.

In [6]:
```
class Person:
    age=0
    def __init__(self,initialAge):
        # Add some more code to run some checks on initialAge
        if initialAge <0:
            print("Age is not valid , setting age to 0")
        else:
            self.age=initialAge
    def amIOld(self):
        # Do some computations in here and print out the correct statement to the console
        if self.age<13:
            print("You are young.")
        elif self.age>=13 and self.age<18:
                print("You are a teenager")
        else:
                print("You are old")
    def yearPasses(self):
        # Increment the age of the person in here
          self.age+=1
t = int(input())
for i in range(0, t):
    age = int(input())
    p = Person(age)
    p.amIOld()
    for j in range(0, 3):
        p.yearPasses()
    p.amIOld()
    print("")
```

```
4
-1
Age is not valid , setting age to 0
You are young.
You are young.

10
You are young.
You are a teenager

16
You are a teenager
You are old

18
You are old
You are old
```

Q2.Problem on Inheritance Task You are given two classes, Person and Student, where Person is the base class and Student is the derived class. Completed code for Person and a declaration for Student are provided for you in the editor. Observe that Student inherits all the properties of Person. Complete the Student class by writing the following: • A Student class constructor, which has 4 parameters: 1. A string, firstName. 2. A string, lastName. 3. An integer, id. 4. An integer array (or vector) of test scores, scores. • A char calculate() method that calculates a Student object's average and returns the grade character representative Of their calculated average The first line contains firstName,lastName and idNumber, separated by a space. The second line contains the number of test scores. The third line of space-separated integers describes scores .

In [ ]:
```
class Person:
    def __init__(self, firstName, lastName, idNumber):
        self.firstName = firstName
        self.lastName = lastName
        self.idNumber = idNumber
    def printPerson(self):
        print("Name:", self.lastName + ",", self.firstName)
        print("ID:", self.idNumber)
class Student(Person):
    def __init__(self, firstName, lastName, idNumber, testScores):
        super().__init__(firstName, lastName, idNumber)
        self.testScores = testScores

    def calculate(self):
        total = 0

        for testScore in self.testScores:
            total += testScore

        avg = total / len(self.testScores)
```

```
        if 90 <= avg <= 100:
            return 'O'
        if 80 <= avg < 90:
            return 'E'
        if 70 <= avg < 80:
            return 'A'
        if 55 <= avg < 70:
            return 'P'
        if 40 <= avg < 55:
            return 'D'
        return 'T'
line = input().split()
firstName = line[0]
lastName = line[1]
idNum = line[2]
numScores = int(input()) # not needed for Python
scores = list( map(int, input().split()) )
s = Student(firstName, lastName, idNum, scores)
s.printPerson()
print("Grade:", s.calculate())
```

In [ ]:

In [ ]:

In [ ]:

In [ ]: