جامعة الانبار
كلية علوم الحاسوب وتكنولوجيا المعلومات
قسم أنظمة شبكات الحاسوب

برمجة كيانية OOP
المرحلة الثانية
الفصل الدراسي الأول والثاني

مدرس المادة
م.د. سميه عبدالله حمد

Creating a List (Array) of Object in Python Class

We can create list of object in Python by appending class **instances** to **list**. By this, every index in the list can point to instance attributes and methods of the class and can access them. If you observe it closely, a list of objects behaves like an array of structures in C.

Example

```
# Python code here creating class
class peoples:
    def __init__(self, name, age):
            self.name = name
            self.age = age
# creating list
list = []
# appending instances to list
list.append( peoples ('Akash', 20) )
list.append( peoples ('Deependra', 40) )
list.append( peoples ('Reaper', 44) )
for obj in list:
    print( obj.name, obj.age, sep =' ' )
# We can also access instances attributes
# as list[0].name, list[0].age and so on.
```

Output:

```
Akash 20
Deependra 40
Reaper 44
```

Example

```
# Python3 code here for creating class
class total:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def Sum(self):
        print( self.x + self.y )
# creating list
list = []
# appending instances to list
list.append( total (2, 3) )
list.append( total (12, 13) )
list.append( total (22, 33) )
for obj in list:
    # calling method
    obj.Sum()
# We can also access instances method
# as list[0].Sum() , list[1].Sum() and so on.
```

Output:

5

25

55

Example

```
#Creating a Python object
class TestDat(object):
    Dat1 = None
    Dat2 = None
#Declaring the Test Array
TestArray = []
#Declaring the object
Test1 = TestDat()
#Defining the member variables in said object
Test1.Dat1 = 0
Test1.Dat2 = 1
#Appending the object to the List
TestArray.append(Test1)
#Rewriting and appending again
Test1.Dat1 = 3
Test1.Dat2 = 4
TestArray.append(Test1)
#Printing our Our Results
print (TestArray[0].Dat1)
print (TestArray[1].Dat2)
```

Output:

3

4

Example

```
class MyList:
    def sublists(self, lst=[]):
        subs = [[]]
        for i in range(len(lst)):
            for j in range (i+1, len(lst)+1):
                sub = lst[i:j]
                subs.append(sub)
        return subs

x = [i for i in range(3)]
my_list = MyList()
print(my_list.sublists(x))
```

Output:

[[], [0], [0, 1], [0, 1, 2], [1], [1, 2], [2]]

Example

```
class list():
    sub = []
    def __init__(self, lst=[]):
        self.sub = lst
a = [ [1,2,3], [4,5,6], [7,8,9], ]
myClass = list(a)
print(myClass.sub)
```

Output:

[[1, 2, 3], [4, 5, 6], [7, 8, 9]]

Example

```
class MyClass(object):
    def __init__(self, number):
        self.number = number
my_objects = []
for i in range(100):
    my_objects.append(MyClass(i))
# later
for obj in my_objects:
    print (obj.number)
```

Output:

0

.

.

99

Example

```
class Bag(object):
    '''This is where the docstring goes.'''
    def __init__(self):
        self.contents = []
    def put_in_bag(self, contents):
        self.contents.append(contents)
    def __str__(self):
        return "The bag has: " + str(self.contents)
if __name__ == "__main__":
    bag1 = Bag()
    bag1.put_in_bag("comb")
```

```
bag1.put_in_bag("candy bar")
print(bag1)
```

Output:

The bag has: ['comb', 'candy bar']

Example: We will take the students' details, (roll number, name, marks in physics, chemistry and maths) from users for multiple students as required by users. And the print the result that displays student's roll number, name and percentage (sum of all marks / 300 * 100).

```
class Student:
    def GetStudentInfo(self):
        self.__rollno = input("Enter Roll Number ")
        self.__name = input("Enter Name ")
        self.__physics = int(input("Enter Physics Marks "))
        self.__chemistry = int(input("Enter Chemistry Marks "))
        self.__maths = int(input("Enter Math Marks "))
    def printResult(self):
        print(self.__rollno,self.__name,((int)((self.__physics
        +self.__chemistry +self.__maths)/300*100 )))
StudentArray = []
while(True):
    student = Student()
    student.GetStudentInfo()
    StudentArray.append(student)
    ch = input("Add More y/n?")
    if(ch=='n'):break
print("Results : ")
for student in StudentArray:
    student.printResult()
```

Output:

```
Enter Roll Number 001
Enter Name John
Enter Physics Marks 87
Enter Chemistry Marks 67
Enter Math Marks 90
Add More y/n?y
Enter Roll Number 002
Enter Name Jane
Enter Physics Marks 54
Enter Chemistry Marks 87
Enter Math Marks 98
Add More y/n?n
Results :
001 John 81
002 Jane 79
```

Example: program to search objects from an array of objects using the filter() method in Python

```
class Student:
  def getStudentInfo(self):
    self.__rollno=input("Enter Roll Number : ")
    self.__name = input("Enter Name : ")
    self.__marks = int(input("Enter Marks : "))
  def printInfo(self):
    print("Roll No. : ", self.__rollno,", Name : ",
self.__name, ", Marks : " , self.__marks)
  def SearchMarks(self,subject,min,max):
```

```
        if(subject=='marks' and (self.__marks >=min and
self.__marks<=max)):
            return True
        else:
            return False
SL=[]
while(True):
    S=Student()
    S.getStudentInfo()
    SL.append(S)
    ch=input("Add More y/n?")
    if(ch=='n'):break
print("All students whose passed the exam are ")
RL=list(filter(lambda S : S.SearchMarks('marks',60,100),SL))
for S in RL:
  S.printInfo()
```

Output:

```
Enter Roll Number : 3
Enter Name : John
Enter Marks : 87
Add More y/n?y
Enter Roll Number : 5
Enter Name : Jane
Enter Marks : 654   45
Add More y/n?y
Enter Roll Number : 1
```

```
Enter Name : Nupur
Enter Marks : 99
Add More y/n?n
All students whose passed the exam are
Roll No. :  3 , Name :  John , Marks :  87
Roll No. :  1 , Name :  Nupur , Marks :  99
```

Example: how to implement a Python program to input and manage students marks list using class and object approach?

```
# Definig a class student, which contain
# name and Roll number and marks of the student
class Student(object):
    def __init__(self, name, roll, marks):
        self.name = name
        self.roll = roll
        self.marks = marks
    def getmarks(self):
        return self.marks
    def getroll(self):
        return self.roll
    def __str__(self):
        return self.name + ' : ' + str(self.getroll()) +'  ::'+
str(self.getmarks())
# Defining a function for building a Record
# which generates list of all the students
     def Markss(rec, name, roll, marks):
    rec.append(Student(name, roll, marks))
    return rec
```

```
# Main Code
Record = []
x = 'y'
while x == 'y':
    name = input('Enter the name of the student: ')
    height = input('Enter the roll number: ')
    roll = input('Marks: ')
    Record = Markss(Record, name, roll, height)
    x = input('another student? y/n: ')

# Printing the list of student
n = 1
for el in Record:
    print(n,'. ', el)
    n = n + 1
```

Output:

```
Enter the name of the student: Prem
Enter the roll number: 101
Marks: 200
another student? y/n: y
Enter the name of the student: Shivang
Enter the roll number: 102
Marks: 250
another student? y/n: y
Enter the name of the student: Radib
Enter the roll number: 103
Marks: 230
```

```
another student? y/n: n
1 .  Prem : 200  ::101
2 .  Shivang : 250  ::102
3 .  Radib : 230  ::103
```