

MSC Degree

Continuous Assessment - 2

CSC 520: Python Programming

1) Subject List & Classlist

- i) Creating a class subject-list.
- ii) Defining the constructor and attributes
- iii) Creating empty lists for name, marks and test-grades.
- iv) Defining a method 'get-input' which has the user input statements.
- v) Getting input from user for the number of students to be graded a 'n'.

n = 2

for i in range(n) # n = 0 1

name = [Peter Parker] i = 0

marks = [10, 20, 30] i = 0

appending to list test grades

i = 1

name = [Dead Pool]

marks = [50, 60, 30]

appending to list test grades.

vi) Creating another class class-list & inheriting the attributes from the base class Subject list.

vii) Defining constructor and attributes

viii) Creating an empty list 'class-list'

ix) Defining method average to find the average of marks.

for i in range(n) # n = 2

using an exception handling if the marks list

is empty

when i = 0

try:

i = 1

avg = sum(testgrades[i][1])/len(testgrades[i][1])

except: ZeroDivisionError

else :

appending average to testgrades.list.

for i in range(n) # n=2

i = 0, 1

if testgrades has an empty marklist
appending an empty list to testgrades
deleting the averages of other marks.

else :

appending test-grades to class list.

Invoking object to class class-list.

calling method average using the object.

2) Difference between Complex numbers & find negative of a number

1) Create a class Operator

2) initialize constructor & define attributes

3) assigning value to attribute

4) Creating a magic function & defining attribute

5) Returning the overloaded object for complex no: calculation
--- Sub ---

6) Creating a magic function --- neg --- & defining attribute 'c' for finding the negative of a no:

7) Returning the overloaded object.

option 1. Diff between 2 complex numbers

2. Neg no:

option = 1

until option > 0 # option = 1

if option == 1 True
option = 2 True
option = 2 False

Creating instance of class &
getting input as complex no:

obj 1 = $i + 3j$

obj 2 = $2 + 1j$

$$\text{obj 1} - \text{obj 2} = \boxed{-1 + 2j}$$

elif option == 2 : option = 2 True

obj 3 = 4

N = obj 3. --neg--() calling the
magic func.
print(N)

elif option > 2 :
option = 3 True
~~# False~~

Invalid choice!

else :

Enter 0 to exit.

3) Meta Characters

- i) Importing library Regular expression.
 - ii) defining a function "is_allowed_specific_char" with an attribute "input_str"
 - iii) Creating a variable 'pattern' and assigning it to a pattern of meta characters.
 - iv) Creating a variable 'result' and assigning it to a library func re.match() for matching the pattern for the input string.
 - v) Getting input from user & assigning it to variable input_str.
 - vi) Calling the function & using input_str as attribute.
-

b) Hat Probabi Probability

4) i) Importing random function

ii) Creating a base class 'Hat'.

iii) Defining the constructor and the attributes.

iv) Defining method Color.

v) Creating an empty list 'A'.

vi) Getting input from user:

a = total no. of balls. ~~to be drawn~~

b = No of balls to be drawn.

while a < b

using an exception handling

try:

if a < b

raise limit error

except limit error

b = enter input again

else:

for i in range(b) # b = 5

i = 0, 1, 2, 3, 4

d = using random values (0, 2)

if d == 0

appending red to list 'A'

if d == 1: count += 1

appending blue to list A

count += 1

if d == 2

appending green to list A.

count += 1

vii) Creating class balls and inheriting attributes of base class 'Hat'

viii) Defining constructor and attributes.

ix) Defining method 'red' & initializing the attributes & calling the attributes of Hat.

x) Defining method blue & attributes of
calling the attributes of Hat.

xi) Defining method green & initializing the
attributes of hat.

xii) Calling the method color inside Hat
in the method green.
