



PyQB

Monga

OO
encapsulationRandom
numbers

Programming in Python¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia
mattia.monga@unimi.it

Academic year 2025/26, I semester

¹© 2025 M. Monga. Creative Commons Attribuzione — Condividi allo stesso modo 4.0 Internazionale. <http://creativecommons.org/licenses/by-sa/4.0/deed.it>

1



PyQB

Monga

OO
encapsulationRandom
numbers

Lecture X: OOP

75



PyQB

Monga

OO
encapsulationRandom
numbers

Object Oriented encapsulation

Encapsulation is so important that it is used also at a higher level: a collection of related procedures.

```
x = 666
```

```
def increment():
    x = x + 1
```

```
def decrement():
    x = x - 1
```

Again: this is correct Python code, but it has problems:

- Both the functions depends on x but this is not clear from their signature: a user must look at the internal details
- The two functions cannot be reused individually, but only together with the other (and x)

76



PyQB

Monga

OO
encapsulationRandom
numbers

Classes

A class is a way to package together a collection of related functions. The class is a “mold” to instance new objects that encapsulated the related functionalities.

```
class Counter:
    def __init__(self, start: int):
        self.x = start
```

```
    def increment(self):
        self.x = self.x + 1
```

```
    def decrement(self):
        self.x = self.x - 1
```

```
c = Counter(666)
c.decrement()
d = Counter(999)
d.increment()
```

77

Random numbers



PyQB

Monga

OO
encapsulation

Random
numbers

Pseudorandomness: the sequence of numbers is not predictable...

```
from random import randint
```

```
# To get a random integer x in the set [1..10]  
x = randint(1, 10)
```

```
from random import randint
```

```
for _ in range(0,10):  
    print(randint(1, 100))
```

unless you know the seed.

```
from random import seed, randint
```

```
seed(292)  
for _ in range(0,10):  
    print(randint(1, 100))
```

78

Exercise



PyQB

Monga

OO
encapsulation

Random
numbers

Write a Python program which chooses an integer 1–10 and asks to the user to guess it

- if the number given by the user is not 1–10, it prints "Invalid";
- if the number is the chosen one, it prints "Yes!";
- otherwise "You didn't guess it..."

Evolve the program: it should now ask until the user guess the number correctly, giving hints ("higher...", "lower...").

How many tries in the worst case? Can you write a program guessing a number between 1 and `int(1e32)`

79

Homework



PyQB

Monga

OO
encapsulation

Random
numbers

- <https://classroom.github.com/a/YYWjMaR2>

80