PES UNIVERSITY

Introduction to Problem solving using generators and iterators

- At the end of this class, students will be able to-
  - Solve Problem solving using generators and iterators

Generators

Generators are your own defined iterators, like **range**.

Generators look like functions, but they keep the state of their variables between calls, and they use **yield** instead of **return**. Also calling them again resumes execution after the **yield** statement.

Generators deal with possibly memory issue as values are generated in the fly.

Example: range(10) returns the numbers between 0 and 9, both inclusive, myrange(10) returns the numbers between 1 and 10.

```
def myrange(number):
    result = 1
    while result <= number:
        yield result
        result += 1

for i in myrange(10):
    print(i)
```

More info: http://www.programiz.com/python-programming/generator

Example: Generating a random gene sequence

```
import random
def randomgene(minlength, maxlength):
    yield 'ATG'
    counter = 2
    while counter < maxlength:
        codon = random.choice('ATCG') + random.choice('ATCG') +
                random.choice('ATCG')
        if codon in ['TGA', 'TAG', 'TAA']:
            if counter >= minlength:
               yield codon
               return
        else:
           yield codon
           counter += 1
    yield random.choice(['TGA', 'TAG', 'TAA'])

# Finally using it
print(''.join(randomgene(40,50)))
```

Example: Generating a random gene sequence, take 2

```
import random
def randomgene(minlength, maxlength):
    if minlength < 2 or minlength > maxlength:
        raise ValueError('Wrong minlength and/or maxlength')
    yield 'ATG'
    stopcodons = ('TGA', 'TAG', 'TAA')
    countdown = random.randrange(minlength, maxlength+1) - 2
    while countdown > 0:
        codon = random.choice('ATCG') + random.choice('ATCG') +
                random.choice('ATCG')
        if codon not in stopcodons:
           yield codon
           countdown -= 1
    yield random.choice(stopcodons)

# Finally using it
print(''.join(randomgene(40,50)))
```