Programming Paradigms

Class Activity #9

[fib_iterator.py] **Q1)**

```
class fibonacci:

	def __init__(self, n):
		self.n = n
		self.i = 0
		self.last_one = 1 # F(n-1)
		self.last_two = 0 # F(n-2)


	def __iter__(self):
		return self


	def __next__(self):
		if self.i < 2 :
			self.i += 1
			return self.i - 1
		if self.i <= self.n:
			self.i += 1
			temp = self.last_two + self.last_one
			self.last_two = self.last_one
			self.last_one = temp
			return self.last_one

		# if we reach here, then i > n
		raise StopIteration()



if __name__ == '__main__':
	for v in fibonacci(3):
		print(v)
```

[fib_generator.py] **Q2) Change the code in Q1 to implement `fibonacci` as a generator.**

```
def fibonacci(n):
	last_one, last_two = 1 , 0
	for i in range(n+1):
		if i < 2:
			yield i
		else:
			temp = last_two + last_one
			last_two = last_one
			last_one = temp
			yield last_one
```