

Lab 15 - Classes

Submission:

This lab has to be demoed on week 13, together with all other labs published between weeks 8 and 13. You will have to explain your code, so make sure it is well documented and you understand it.

Objective:

To practice, understand and design Python classes.

Questions:

1. Try to answer this question before running the code. You might run it to check your answer.

```
class MyClass (object):
    def method1(self, param_list):
        self.local_list = []
        for element in param_list:
            if element > 10:
                self.local_list.append(element)

    def method2(self):
        self.sum_int = 0
        for element in self.local_list:
            self.sum_int += element
        return self.sum_int
```

```
inst1 = MyClass()
inst2 = MyClass()
inst1.method1([1,2,3])
print(inst1.local_list) # Line 1
inst1.method1([10,11,12])
print(inst1.local_list) # Line 2
print(inst1.method2()) # Line 3
#inst2.method2() # Line 4
```

- (a) What output is produced by # Line 1 of the above program?
- (b) What output is produced by # Line 2 of the above program?
- (c) What output is produced by # Line 3 of the above program?
- (d) # Line 4 is commented out. What result would occur if Line 4 were executed by the program. Why?

2. Try to answer this question before running the code. You might run it to check your answer.

```
class NewClass(object):
    def __init__(self, param_int=1):
        self.the_int = param_int
        if param_int % 2 == 0:
            self.parity = 'even'
        else:
            self.parity = 'odd'

    def process(self, instance):
        sum_int = self.the_int + instance.the_int
        if sum_int < 0:
            return 'negative'
        elif sum_int % 2 == 0:
            return 'even'
        else:
            return 'odd'

    def __str__(self):
        return 'Value {} is {}'.format(self.the_int, self.parity)

inst1 = NewClass(4)
inst2 = NewClass(-5)
inst3 = NewClass()
print(inst1) # Line 1
print(inst1.parity) # Line 2
print(inst1.process(inst2)) # Line 3
print(inst3.process(inst1)) # Line 4
```

- (a) What output is produced by Line 1 of the example program?
- (b) What output is produced by Line 2 of the example program?
- (c) What output is produced by Line 3 of the example program?
- (d) What output is produced by Line 4 of the example program?

3. Sample class describing a Person

```

class Person():
    def __init__(self, fname, sname, address):
        self.f_name = fname
        self.s_name = sname
        self.address = address

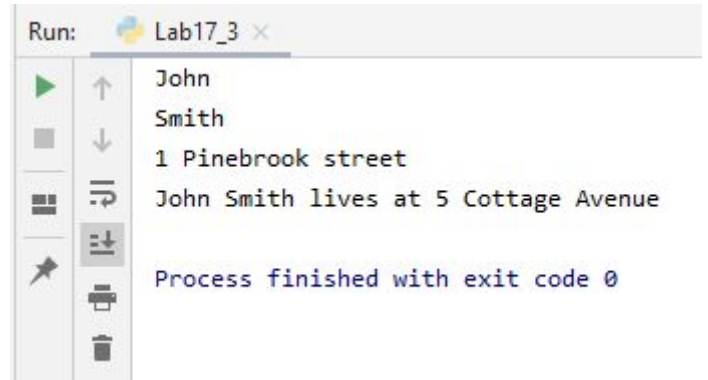
    def change_address(self, new_address):
        self.address = new_address

    def __str__(self):
        return self.f_name + " " + self.s_name + " lives at " + self.address

# main
p1 = Person("John", "Smith", "1
Pinebrook street")
print(p1.f_name)
print(p1.s_name)
print(p1.address)

p1.change_address("5 Cottage Avenue")
print(p1)

```



Run: Lab17_3 x

```

John
Smith
1 Pinebrook street
John Smith lives at 5 Cottage Avenue

Process finished with exit code 0

```

(a): Design a class to represent a **rectangle**. Some methods examples can be the rectangle area and rectangle perimeter.

(b): Design a class to represent a **bank account**. Some information you might want in a bank account are the IBAN, account number, available funds, a list with the last 5 transactions. You might also add methods to withdraw and deposit money.