

## Python Sample Solutions - Lab 3

# Note there is likely more than one way to do every exercise.

# Below are some sample solutions

#Exercise 1.

class Rectangle:

'Class Rectangle'

def \_\_init\_\_(self, length, width):

self.length = length

self.width = width

def perimeter(self):

return 2\*(self.width + self.length)

def area(self):

return self.width \* self.length

r1 = Rectangle(2, 3)

print r1.perimeter()

print r1.area()

#Exercise 2.

class BankAccount:

'Class BankAccount'

def \_\_init\_\_(self, balance = 0):

self.balance = balance

def withdraw(self, amount):

self.balance = self.balance - amount

def deposit(self, amount):

self.balance = self.balance + amount

def get\_balance(self):

return self.balance

```
ba = BankAccount(100)
print str(ba.get_balance())
ba.deposit(5000)
ba.withdraw(10000)
print str(ba.get_balance())
```

#Exercise 3 and Exercise 4.

```
class Person:
    'Class Person'

    def __init__(self, name, birth_year):
        self._name = name
        self.birth_year = birth_year

    def get_age(self):
        return 2014 - self.birth_year

    def get_name(self):
        return self._name

    def get_details(self):
        return 'Hello, I am a Person'
```

```
class Student(Person):
    'Class Student'

    def __init__(self, name, birth_year, student_number):
        Person.__init__(self, name, birth_year)
        self.student_number = student_number

    def get_details(self):
        return self.student_number

    def mature(self):
        return (self.get_age() > 23)
```

```
s = Student('Alice', 1992, 12345)
print str(s.get_details())
print s.get_name()
print s.mature()
```

```
#Exercise 5.  
nums = [1, 2, -2, -3, 5, 4]
```

```
for n in nums:  
    if n < 0:  
        print n
```

```
#Exercise 6.  
sq = lambda arg1: 2 ** arg1  
print sq(3)
```

```
#Exercise 7.  
def even(n):  
    for x in range(2, n+1):  
        if (x % 2 == 0) or (x % 3 == 0):  
            print x
```

```
even(17)
```

```
#Exercise 8.  
def month(n):  
    months = {1 : 'Jan', 2 : 'Feb', 3 : 'Mar', 4 : 'Apr', 5 : 'May', 6 : 'Jun',  
7 : 'Jul', 8 : 'Aug', 9 : 'Sep', 10 : 'Oct', 11 : 'Nov', 12 : 'Dec'}  
    return months[n]
```

```
print month(8)
```

```
#Exercise 9.  
def valid_isbn(isbn):  
  
    if len(isbn) == 13 and isbn.count('-') == 3:  
        return True  
    return False
```

```
print valid_isbn('0-123-12345-0')
```

```
#Exercise 10.
name = raw_input("Please enter your name: ")
student_num = raw_input("Please enter your student number: ")

ask = True
subjects = []
while ask:
    subject = raw_input("Please enter a subject. Enter 'done' when
finished: ")
    if subject == 'done':
        ask = False
        break
    subjects.append(subject)

print name
print student_num
print subjects
```

```
#Exercise 11.
a = [3, 5, 1, 7, 9]
b = [4, 2, 6, 3, 9]

def intersect(list1, list2):
    return list(set(list1) & set(list2))

print intersect(a, b)

#or

def intersect(list1, list2):
    return set(list1).intersection(list2)

print intersect(a, b)
```

#Exercise 12.

```
def pay(wage, hours):  
    time_and_half = wage + wage/2  
    if hours > 40:  
        overtime = hours - 40  
        return (40 * wage) + (overtime * time_and_half)  
    else:  
        return wage * hours  
  
print pay(50, 50)
```

#Exercise 13.

```
grades = [[95, 92, 86, 87], [66, 54], [89, 72, 100], [33, 0, 0]]
```

```
def avg(list):  
    for l in list:  
        print (sum(l) / len(l))
```

```
avg(grades)
```

#Exercise 14.

```
choice = raw_input('Enter C for celsius-fahrenheit conversion or F for  
fahrenheit-celcius conversion: ')
```

```
if choice == 'C' or choice == 'c':  
    celsius = int(raw_input('Enter degree Celsius: '))  
    fahrenheit = ((celsius * 9)/5) + 32  
    print fahrenheit  
elif choice == 'F' or choice == 'f':  
    fahrenheit = int(raw_input('Enter degree fahrenheit: '))  
    celsius = ((fahrenheit - 32) * 5) / 9  
    print celsius  
else:  
    print 'Invalid choice. Exit'
```

#Exercise 15.

try:

    # Open

    my\_file = open("output.txt", "r")

    # Read

    print my\_file.read()

    # Close

    my\_file.close()

except IOError, e:

    print 'catching IO exception'