Higher Order Function and Lambda Calculus

Question 1.

Run the following code, observe the displayed result.

```
def add (x,y):
    return x+y

def inc(f,x):
    return f(x,1)

print(inc(add,2))
```

Answer: 3

Question 2.

Rewrite inc(f,x) using lambda calculus

Question 3.

Develop the following functions using the similar manner.

a. add10(x)

b. dec(f,x)

Question 4.

Run the following code, observe the displayed result.

```
def create_adder(x):
    def adder(y):
        return x + y

    return adder

add_15 = create_adder(15)

print(add_15(10))
```

Question 5.

Rewrite create_adder using lambda

Question 6.

a. Develop create_lifter(x) to return x^y

b. Use create_lifter to develop area(n), which returns the area of a square whose side length is n.

List comprehension

Question 7.

Run the following code and observe the result

a.

```
def rem(l,i):
   return [x for x in l if x !=i]
print(rem([2,3,3,4],3))
```

b.

```
def rem(l,i):
   return [x if x!=i else -1 for x in l]

print(rem([2,3,3,4],3))
```

Question 8.

Using list comprehension to accomplish the following functions

a. Extract list of even numbers from a list

b. Return a squared list of a list

c. Remove all of square numbers less than 100 in a list

Using map

Question 9.

Run the following code

```
def f(x):
    return x+1

l = list(map(f,[2,3,4]))
print(l)
```

Question 10.

Rewrite Question 9 using lambda expression

Answer:

```
l = list(map(lambda x:x+1,[2,3,4]))
print(l)
```

Question 11.

Use map and lambda to convert a list of numbers into the squared list.

Customized class

Question 12.

Run the following code

```
class C:
    #n:int
    def __init__(self, k):
        self.n = k

    def __str__(self):
        return str(self.n)

print(C(3))
```

Question 13.

Convert a list of numbers into a list of C objects using *map*

```
class C:
    #n:int
    def __init__(self, k):
        self.n = k

    def __str__(self):
        return str(self.n)

def int2C(x):
    return C(x)

def convert(l):
    return list(map(int2C,l))

list_C = convert([2,3,4])
print(list_C)

print(" ".join(str(i) for i in list_C))
```

Question 14.

Rewrite Question 13 using lambda

Question 15.

Declare a class Parent whose children are of class C as follows. Run the code and observe the result.

```
class Parent:
    #decl:list(C)
    def __init__(self, decl):
        self.decl = decl

    def __str__(self):
        return "Parent([C" + ',C'.join(str(i) for i in self.decl) + "])"
```

```
class C:
    #n:int
    def __init__(self, k):
        self.n = k

    def __str__(self):
        return str(self.n)

def convert(l):
    return list(map(lambda x: C(x),l))

list_C = convert([2,3,4])

p = Parent(list_C)

print(p)
```

Question 16.

Write a function using map and lambda to convert a list of number into the corresponding Parent object.