

### **Practical no 10 : Demonstrate implementation of the Anonymous Function Lambda.**

A

```
# Function definition is here
sum = lambda arg1, arg2: arg1 + arg2;
# Now you can call sum as a function
print "Value of total : ", sum( 10, 20 )
print "Value of total : ", sum( 20, 20 )
```

B

```
# Python code to illustrate cube of a number
# showing difference between def() and lambda().
def cube(y):
    return y*y*y
lambda_cube = lambda y: y*y*y
# using the normally
# defined function
print(cube(5))
# using the lambda function
print(lambda_cube(5))
```

### **Practical 11: Demonstrate implementation Mapping Functions over Sequences.**

```
def mul(i):
    return i * i
num = (3, 5, 7, 11, 13)
resu = map(mul, num)
print(resu)
# making the map object readable
mul_output = list(resu)
print(mul_output)
```

**Practical 12: Demonstrate implementation functional programming tools such as filter and reduce.**

A

```
scores = [66, 90, 68, 59, 76, 60, 88, 74, 81, 65]
def is_A_student(score):
    return score > 75
over_75 = list(filter(is_A_student, scores))
print(over_75)
```

B

```
dromes = ("demigod", "rewire", "madam", "freer", "anutforajaroftuna", "kiosk")
palindromes = list(filter(lambda word: word == word[::-1], dromes))
print(palindromes)
```

C

# Python 3

```
from functools import reduce
numbers = [3, 4, 6, 9, 34, 12]
def custom_sum(first, second):
    return first + second
result = reduce(custom_sum, numbers)
print(result)
```

D

```
from functools import reduce
numbers = [3, 4, 6, 9, 34, 12]
def custom_sum(first, second):
    return first + second
result = reduce(custom_sum, numbers, 10)
print(result)
```

**Practical 13 :** Demonstrate the Module Creation, Module usage, Module Namespaces, Reloading Modules, Module Packages, Data Hiding in Modules.

### **A Module Creation**

```
def add(a, b):  
    """This program adds two  
    numbers and return the result"""  
    result = a + b  
    return result
```

#### **import module**

```
import math  
print("The value of pi is", math.pi)
```

#### **Import with renaming**

```
import math as m  
print("The value of pi is", m.pi)
```

#### **from...import statement**

```
from math import pi  
print("The value of pi is", pi)  
  
Run Code
```

#### **Import all names**

```
from math import *  
print("The value of pi is", pi)
```