CSE 2040 Programming IV Lecture #24

What will we learn about functions

- Function as parameter
- Returning a function
- Higher order functions

Function as a parameter – Another example

```
def getStudentList():
    ''' Gets list of students from database '''
    studentList = ["X1", "X2", "X3", "X4"]
    return studentList
def getCourseList():
    ''' Gets list of courses from database '''
    courseList = ["PIII", "DM", "CO", "EIV"]
    return courseList
def printList(fetchList):
    list = fetchList()
    for item in range(len(list)):
        print(list[item])
# In main
printList(getStudentList)
printList(getCourseList)
```

Returning a function - Example

```
def polynomial_generator(a, b, c):
    def polynomial(x):
        return a * x**2 + b * x + c
    return polynomial

poly1 = polynomial_generator(2, 3, -1)
poly2 = polynomial_generator(-1, 2, 1)

print()
print("Polynomial of degree 2 ... ")
print(poly1(-2))
print(poly2(-2))
```

1000

poly1

poly2

```

Polynomial of degree 2 ...
1
-7
>>>
```

Higher Order Functions

```
def sumItUp(n, term):
        sum = 0
        ith = 1
        while ith <= n:
            sum = sum + term(ith)
            ith = ith + 1
        return sum
```

Abstracted common code of similar summation functions into a separate function

```
def naturalTerm(x):
        return x
def cubeTerm(x):
        return x * x * x;
def series1Term(x):
        return 1/x
```

Specific functions to provide ONLY the difference in the code

```
sumItUp(n, naturalTerm)
sumItUp(n, cubeTerm)
sumItUp(n, series1Term)
```

Calling the abstracted function

About functions

- Function identifier can be assigned to another identifier
- Function can be defined inside another function
- Parameter to a function and return value from a function
  - Function can take a non-function as a parameter and return a non-function
  - *Function can take a non-function as a parameter and return a function*
  - *Function can take a function as a parameter and return a non-function*
  - *Function can take a function as a parameter and return a function*

Higher-order functions

References

- Reference from Professor Usha Slides
- Dr. R. Nageswara Rao, Core Python Programming, Second Edition, 2018
    - Page - 248 (Program 11 and 12 and 13)