Notes & Summary For Section 3

Section 4, Lecture 34

**Notes For Section 3**

**Returning a value from a function:**  
The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.All the above examples are not returning any value.

You can return a value from a function as follows −  
   
 # Function definition is here  
def sum( arg1, arg2 ):  
   # Add both the parameters and return them."  
   total = arg1 + arg2  
   print "Inside the function : ", total  
   return total;

# Now you can call sum function  
total = sum( 10, 20 );  
print "Outside the function : ", total

**Passing functional arguments to functions:**

We can also pass function as an argument to other function example:

def add(a,b):  
    return a + b  
def square(c):  
    return c \* c

square(add(2,3))

**Modules in Python:**  
A module allows you to logically organize your Python code. Grouping related code into a module   
makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.Simply, a module is a file consisting of Python code. A module can define functions, classes and variables.  A module can also include runnable code. You can use any Python source file as a module by executing an import statement in some other Python source file.

 example:  
 import module\_name  
  Here module\_name is the name of the module which contains the code which you want to use.