1. Write a Python program that prints all the numbers from 0 to 9 except 3,6 and 9
2. Write a Python program that accepts a word from the user and reverse it.
3. Write a Python program which iterates the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
4. Write a Python program to check whether an alphabet is a vowel or consonant.
5. Write a script to print out the first 20 Fibonacci numbers. You can use the internet to find the definition of the Fibonacci sequence if you need to.
6. Create three lists, each of which can hold 3 values:
   1. the first list should contain 3 names
   2. the second list should contain the ages of the people with the names above
   3. the third list should contain the email addresses

Using the following construct, enter values into the three lists:

for counter in range(0,3):

Now print out each of the lists.

1. Write a function that takes named parameters and returns the parameters as a Dictionary of parameter name -> value
2. Rewrite your Fibonacci exercise from earlier to use a function that takes an argument for how many numbers are required:

Test with :fib(2000),fib(100),fib(0)

Rewrite the function again, so that it returns a list of values.

1. Create a class that represents a car. A car should have the following attributes:
   1. speed
   2. top speed
   3. registration (license plate) number

Add methods to provide access to the attributes. Also add a method accelerate, which will throw an exception if the car attempts to exceed its top speed.

1. Design and implement a small set of classes to represent a set of Bank Accounts.
   1. There should be two types of accounts; a cheque account that does not pay interest but allows a configurable overdraft, and a savings account that pays interest but does not allow overdrafts.
   2. An account should be identifiable by an account number, and the account holder's name should also be available.
   3. Decide which information needs to be in the base class (and whether this has to be abstract or not).
2. For your Bank Account classes from [the earlier lab](http://wiki.ms.com/MallonAssociates/AdvPython-InheritanceLab), refactor so that the overdraft limit of a cheque account is accessible as an attribute rather than a simple property.
3. Write a script that calculates the factorial of an integer in the range 0 to 10 (0! is the same as 1!).

Now add code to verify that the number is indeed in the range 0 to 10, and throw an exception if it is greater than 10, or less than 0.

1. Write a Python script that accepts input from the user until the user types "end".

Append all the information into a file. Close and reopen the same file in read mode and print out the history of what the user typed.

Remember to use a try block/with to cleanly close your file resources.

1. Show how to verify that a line of input
   1. contains only letters
   2. contains only letters, but if not show the first character on the line that is not a letter.
   3. contains only letters, but if not show all the non-letter characters (without using a loop).
2. Construct a regular expression, to use with the split() function, that allows you to find all the individual words in a line.

When used with the line

Twinkle, twinkle, little star!

it should return an array containing the words Twinkle, twinkle, little and star.

1. In the string

Do dogs eat cats, or does your cat eat a mouse?

perform a replace of dog to cat, cat to mouse, and mouse to dog.

1. We have a nested loop. In the inner loop we are using continue for the third iteration. On the third iteration compiler will skip the 3rd iteration and move to 4th iteration in the same loop. WAP so that the fourth iteration starts with the parent loop
2. WAP to get a number from user and check if the number is Armstrong number or not?

Use a conventional method and also use lambda and map function to find the number,

Eg. 153 – The sum of 1^3+5^3+3^3 🡺 1+125+27 = 153, so 153 is an Armstrong number.

1. WAP to return the square of a given number. Let’s say the count of numbers is 1 million. In that case there will be performance issue. In that case what approach will be followed?

HINT: Generators

1. WAP to sort the given student tuple based on age and department in descending order.

A, B, b, C are departments

student\_tuples = [

('John', 'A', 15),

('Jane', 'B', 12),

('Dave', 'C', 10),

('Dave', 'b', 10)]

1. Using list comprehension, create list of first letter of each word from given list.

listOfWords = ["this","is","a","list","of","words"]

The output should be: ['t', 'i', 'a', 'l', 'o', 'w']

1. WAP to create dictionary with key as number and values as cube of number, for numbers in range 10.
2. WAP to parse data from text file and then print the data in a dictionary.

Address: 1423 99th Ave

Square feet: 1705

Price\_usd: 340210

1. Please find below a XML tag:

<?xml version="1.0"?>

<catalog>

<book id="bk101">

<author>Gambardella, Matthew</author>

<title>XML Developer's Guide</title>

    <genre>Computer</genre>

     <price>44.95</price>

     <publish\_date>2000-10-01</publish\_date>

     <description>An in-depth look at creating applications

     with XML.</description>

   <book id="bk102">

     <author>Ralls, Kim</author>

     <title>Midnight Rain</title>

<genre>Drama</genre>

     <price>23.90</price>

     <publish\_date>1990-08-21</publish\_date>

     <description>A melodrama at its best.</description>

</catalog>