Python for Tool Developers

# Lab 3

# 11/11/2016

### Lab 3. Iterators and Generators

One possible task you may be asked to do is to analyze e-mails for suspicious activity. However, e-mail applications generate very large amounts of data, For any given user, the amount of e-mail generated may be in excess of a gigabyte or more, depending on history and usage. In this lab, we will use iterators and generators to help us write a simple application that will store potentially suspicious e-mails.

You will create a class called *mail\_list*. This class will read a database of e-mails and store relevant e-mails in a dictionary attribute inside the class.

Since the database is rather large, it is unrealistic to be able to expect to bring the entire file into memory, instead a generator will be created that will only read in e-mails from the file as required.

Additionally, only the e-mails that meet a specific keyword criteria will be retained in the mail\_list class and stored. The mail\_list class will need to be an iterable object so that a user of the class can iterate over the e-mails with a for loop.

Exercises

1. Create the mail\_list class.

2. Create a generator which will be used to store the e-mails into a dictionary attribute

based on some keyword criteria given in the initializer. As the files are stored in a tarred, gzip'ed format, consider using the *tarfile* library module provided by Python.

3. Store the e-mails in a dictionary based on a key = sender and value is a list of mail bodies.

3. Make the mail\_list class iterable.

4. Allow users to query the number of e-mails that match the keyword criteria and the sender.