**Directed Reading**

**Study and Development of Call Graph for Open Source Projects**

**Week -3**

**Name: Megha Nagabhushan**

**Student ID: 16226858**

**Starting off with pycallgraphs**

The Pycallgraph application generates call graphs for one python file. Since the open source projects contain multiple modules which are independent, the existing application has to be enhanced to accommodate the requirement.

**Alternative approach – using inspect module provided by Python**

Alternative approach is to use methods provided by python library to know the incoming and outgoing method calls for a particular method.

The [inspect](https://docs.python.org/3/library/inspect.html#module-inspect) module provides several useful functions to help get information about live objects such as modules, classes, methods, functions, tracebacks, frame objects, and code objects.

Inspect helps in –

* examining the contents of a class
* retrieving the source code of a method
* getting all the information you need to display a detailed traceback

**Looking for the incoming calls to a method.**

curframe = inspect.currentframe()

calframe = inspect.getouterframes(curframe, 2)

print 'caller name:', calframe[1][3]

currentframe() -> Return the frame object for the caller’s stack frame.

Getouterframes() -> A list of [named tuples](https://docs.python.org/3/glossary.html#term-named-tuple) FrameInfo(frame, filename, lineno, function, code\_context, index)  is returned.

**Looking for outgoing calls from a method.**

curframe = inspect.currentframe()

calframe = inspect.getinnerframes(curframe, 2)

print ‘calling method name:’, calframe[0][3]

Reference: <https://docs.python.org/3/library/inspect.html#inspect.getframeinfo>

Tasks for the following week:

* use getouterframes() and getinnerframes() for all the methods in an open source project.
* Log the results returned by these methods.