How to create seperate log file Based on Caller:

```
import logging
import inspect
def getCustomLogger(level):
    loggername=inspect.stack()[1][3]
    logger=logging.getLogger(loggername)
    logger.setLevel(level)

    fileHandler=logging.FileHandler('{}.log'.format(loggername),mode='a')
    fileHandler.setLevel(level)

    formatter = logging.Formatter('%(asctime)s - %(name)s - 
 %(levelname)s: %(message)s',datefmt='%m/%d/%Y %I:%M:%S %p')
    fileHandler.setFormatter(formatter)
    logger.addHandler(fileHandler)

    return logger
```

test.py:

#Same as previous

```
import logging
from customlogger import getCustomLogger
class LoggingDemo:
    def m1(self):
        logger=getCustomLogger(logging.DEBUG)
        logger.debug('m1:debug message')
        logger.info('m1:info message')
        logger.warn('m1:warn message')
        logger.error('m1:error message')
        logger.critical('m1:critical message')
    def m2(self):
        logger=getCustomLogger(logging.WARNING)
        logger.debug('m2:debug message')
        logger.info('m2:info message')
        logger.warn('m2:warn message')
        logger.error('m2:error message')
        logger.critical('m2:critical message')
    def m3(self):
        logger=getCustomLogger(logging.ERROR)
        logger.debug('m3:debug message')
        logger.info('m3:info message')
        logger.warn('m3:warn message')
        logger.error('m3:error message')
        logger.critical('m3:critical message')

```