```
SCons: A software construction tool
- Configuration files are Python scripts.
- Built-in support for C, C++, D, Java, Fortran, Yacc, Lex, Qt and SWIG,
and building TeX and LaTeX documents.
- Cofiguration file name should be SConstruct
- To run use command scons


Building Simple C / C++ Programs:
      Program('hello.c')

Specifying the Name of the Target:
      Program('new_hello', 'hello.c')

Building Object Files:
      Object('hello.c')

Compiling Multiple Source Files:
           Program('program', ['prog.c', 'file1.c', 'file2.c'])

Building Libraries:
           Library('foo', ['f1.c', 'f2.c', 'f3.c'])
        StaticLibrary('foo', ['f1.c', 'f2.c', 'f3.c'])
        SharedLibrary('foo', ['f1.c', 'f2.c', 'f3.c'])

Linking with Libraries:

      Library('foo', ['f1.c', 'f2.c', 'f3.c'])
      Program('prog.c', LIBS=['foo', 'bar'], LIBPATH='.')

Implicit Dependencies:
       Program('hello.c', CPPPATH = '.')

Creating a Construction Environment:

      import os
        env = Environment()
        env = Environment(CC = 'gcc',
                          CCFLAGS = '-O2')
        env.Program('foo.c')


Example of DM:

import os
env = Environment()
env.Append(IOTVITY_DIR ='/home/vankagan/Ganesh/iot/iotivity-1.2.1')

iotvity_dir = env.get('IOTVITY_DIR')

env.AppendUnique(CXXFLAGS = ['-std=c++0x', '-Wall'])

env.SharedLibrary('DM_iotivity','DM_iotivity.cpp', CPPPATH = [iotvity_dir
+ '/resource/include/',
           iotvity_dir + '/resource/csdk/stack/include',
           iotvity_dir + '/resource/c_common/ocrandom/include',
           iotvity_dir + '/resource/c_common/',
                 iotvity_dir + '/resource/csdk/logger/include',
           iotvity_dir + '/resource/oc_logger/include',
           iotvity_dir + '/resource/examples',
```

```
            './'
            ],
        LIBS = ['coap', 'connectivity_abstraction' , 'oc_logger' ,
'octbstack' , 'oc'],
        LIBPATH= [iotvity_dir + '/out/linux/x86_64/release'])
```