**COSC 354.001**

**Input – Preventing a Deadlock C++**

#include <iostream>

#include <boost/thread/thread.hpp>

#include <boost/thread/mutex.hpp>

boost::mutex mutex1, mutex2;

void ThreadA()

{

// Creates deadlock problem

mutex2.lock();

std::cout << "Thread A" << std::endl;

mutex1.lock();

mutex2.unlock();

mutex1.unlock();

}

void ThreadB()

{

// Creates deadlock problem

mutex1.lock();

std::cout << "Thread B" << std::endl;

mutex2.lock();

mutex1.unlock();

mutex2.unlock();

}

void ExecuteThreads()

{

boost::thread t1( ThreadA );

boost::thread t2( ThreadB );

t1.join();

t2.join();

std::cout << "Finished" << std::endl;

}

int main()

{

ExecuteThreads();

return 0;

}

Output

Thread B Thread A