Functionality of mininet_custom.py:

        This file essentially launches a virtual network capable of supporting SDN/Pox. It uses various methods from the "mininet.net" module to spawn virtual hosts and switches, and then connects them using the "addLink" method. We modified the original code provided to match the topology specified in the lab 6 writeup.

Functionality of lab6_controller.py:

        This code is a modified version of the forwarding l2_pairs.py example controller. The launch function invokes the packetIn function whenever a "Packet In" flow is sent to the controller. In order to support the blocking of some specific two way communication, we created a dictionary that keeps track of which hosts are allowed to speak to each other. Once the "Packet In" flow arrives, the function parses the packet information, and then learns the source by adding it to a separate dictionary called 'table'. Then, it invokes the 'get' method on the dictionary to determine whether or not the port is known. If the destination port is unknown (the 'get' method returns 'None'), it does a mock broadcast by sending out ARP requests in order to determine where the destination host is. If the destination port is known, we first check to make sure that the source and destination are allowed to talk to each other. If that's the case, we first create a message that gives forwarding rules to both the source and destination and send it out on the event port. This way, after a flow is analyzed for the first time, the switch no longer has to reach out to the controller and can just forward based on the rules provided. After that, we want to forward the original packet to the proper destination based on the forwarding rules established previously. If the two hosts attempting to communicate are not allowed to, the controller simply returns nothing to the host which is an implied "drop the packet." Now we have a simple learning bridge implementation with the ability to block traffic between certain hosts. However, it is not a pure solution because initial ARP traffic can still travel between all the hosts, regardless of who can/can't talk to each other.