Appendix A: Educational Material (Paxos Algorithm)

Pseudocode for the Paxos Algorithm

The Paxos algorithm is easily described through pseudocode.
Source: http://nil.csail.mit.edu/6.824/2015/notes/paxos-code.html

```
--- Paxos Proposer ---

1    proposer(v):
2    while not decided:
3        choose n, unique and higher than any n seen so far
4        send prepare(n) to all servers including self
4        if prepare_ok(n, n_a, v_a) from majority:
5          v' = v_a with highest n_a; choose own v otherwise
6          send accept(n, v') to all
7          if accept_ok(n) from majority:
8            send decided(v') to all


--- Paxos Acceptor ---

9    acceptor state on each node (persistent):
10    n_p     --- highest prepare seen
11    n_a, v_a --- highest accept


12   acceptor's prepare(n) handler:
13    if n > n_p
14      n_p = n
15      reply prepare_ok(n, n_a, v_a)
16   else
17     reply prepare_reject
18   acceptor's accept(n, v) handler:
19    if n >= n_p
20      n_p = n
21      n_a = n
22      v_a = v
23      reply accept_ok(n)
24   else
25     reply accept_reject
```

```
--- Paxos Learner ---

26   learner learn(n, v) handler:
27        add (n, v) to consensus list.
28        if acceptor value was learned:
29             reset acceptor number and value.
30             n_a = nil
31             v_a = nil
```

Explanation

A Paxos round begins with the proposer. A leader is selected and executes the proposer sequence to propose a new consensus value. The goal of the first step is for the leader to gain permission to propose a value to the acceptors. First, it proposes a prepare request to a quorum of the acceptor nodes. If this is the highest sequence number (`n` in the above pseudocode) the acceptor has seen so far, it replies with a promise to not accept any further proposals with sequence numbers lower than the current proposer's sequence number. If the acceptor has previously accepted a value, it can append this proposal number and proposal value to its response. Alternatively, if the acceptor has already seen a prepare with a higher sequence number, it simply replies with a reject. Once the proposer receives a quorum of the acceptors, it may continue with the proposal process. Otherwise, it fails to propose its value this round and tries again next round.

In the next phase (the accept phase), now that the leader has gained permission, it proposes a proposal value to all of the acceptors. This proposal value is the highest ranked accepted value sent in the prepare replies from the acceptors in the previous step, or its own value if all of the acceptor reply values are nil. If the receiving acceptors have not promised to reject this proposal, it must accept this proposal value and update its accepted value, proposal number, and promised number. Once the proposer receives a quorum of the acceptors again, it may finish the proposal process. It can broadcast that its chosen value to the rest of the server nodes to indicate that consensus has been reached. When servers receive this request, their learner handler adds the consensus value to its consensus list and updates resets its acceptor value if necessary.