ANSWER

When you download the handout, you should see the following files:

- `a.js`
- `a.sh` (to be used by Mac/Linux users)
- `a.bat` (to be used by Windows users)

I will be performing this on a mac in the `~/Downloads` directory, but your solution should be similar.

```
./a.sh
```

giving the output of:

```
Already running mongo* processes (this is fyi, should be none probably):
50275 ttys002    0:00.00 grep mongo

make / reset dirs

running mongod processes...
about to fork child process, waiting until server is ready for connections.
forked process: 50282
child process started successfully, parent exiting
about to fork child process, waiting until server is ready for connections.
forked process: 50285
child process started successfully, parent exiting
about to fork child process, waiting until server is ready for connections.
forked process: 50288
child process started successfully, parent exiting

50282 ??         0:00.07 mongod --fork --logpath a.log --smallfiles --oplogSize 50 --p
ort 27001 --dbpath data/z1 --replSet z
50285 ??         0:00.05 mongod --fork --logpath b.log --smallfiles --oplogSize 50 --p
ort 27002 --dbpath data/z2 --replSet z
50288 ??         0:00.06 mongod --fork --logpath c.log --smallfiles --oplogSize 50 --p
ort 27003 --dbpath data/z3 --replSet z
50291 ttys002    0:00.00 grep mongo

Now run:

  mongo --shell --port 27003 a.js
```

So I do as it asks:

```
mongo --shell --port 27003 a.js
```

```
ourinit()
```

```
{
        "info" : "Config now saved locally.  Should come online in about a minute.",
        "ok" : 1
}
waiting for set to initiate...
{
        "setName" : "z",
        "setVersion" : 1,
        "ismaster" : false,
        "secondary" : true,
        "hosts" : [
                "localhost:27003",
                "localhost:27001"
        ],
        "arbiters" : [
                "localhost:27002"
        ],
        "me" : "localhost:27003",
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 1000,
        "localTime" : ISODate("2015-05-01T20:02:22.656Z"),
        "maxWireVersion" : 2,
        "minWireVersion" : 0,
        "ok" : 1
}
ok, this member is online now; that doesn't mean all members are
ready yet though.
```

```
rs.status()
```

```
{
        "set" : "z",
        "date" : ISODate("2015-05-01T20:08:49Z"),
        "myState" : 1,
        "members" : [
                {
                        "_id" : 1,
                        "name" : "localhost:27001",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 386,
                        "optime" : Timestamp(1430510540, 1),
                        "optimeDate" : ISODate("2015-05-01T20:02:20Z"),
                        "lastHeartbeat" : ISODate("2015-05-01T20:08:48Z"),
                        "lastHeartbeatRecv" : ISODate("2015-05-01T20:08:48Z"),
                        "pingMs" : 0,
                        "syncingTo" : "localhost:27003"
                },
                {
                        "_id" : 2,
                        "name" : "localhost:27002",
                        "health" : 1,
                        "state" : 7,
                        "stateStr" : "ARBITER",
                        "uptime" : 386,
                        "lastHeartbeat" : ISODate("2015-05-01T20:08:48Z"),
                        "lastHeartbeatRecv" : ISODate("2015-05-01T20:08:49Z"),
                        "pingMs" : 0
                },
                {
                        "_id" : 3,
                        "name" : "localhost:27003",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 416,
                        "optime" : Timestamp(1430510540, 1),
                        "optimeDate" : ISODate("2015-05-01T20:02:20Z"),
                        "electionTime" : Timestamp(1430510551, 1),
                        "electionDate" : ISODate("2015-05-01T20:02:31Z"),
                        "self" : true
                }
        ],
        "ok" : 1
}
```

At this point, I've shut down the mongod at port 27001. Also, my mongod at port 27003 is primary, so I'm good to go; otherwise, I might have needed to do the following:

```
z:SECONDARY> exit
bye
```

```
mongo --port 27001
```

```
z:PRIMARY> rs.stepDown()
2015-05-01T16:23:04.511-0400 DBClientCursor::init call() failed
```

```
2015-05-01T16:23:04.513-0400 Error: error doing query: failed at src/mongo/shell/query
.js:81
2015-05-01T16:23:04.514-0400 trying reconnect to 127.0.0.1:27001 (127.0.0.1) failed
2015-05-01T16:23:04.515-0400 reconnect 127.0.0.1:27001 (127.0.0.1) ok
z:SECONDARY>exit
bye
```

```
mongo --shell --port 27003 a.js
```

Whether I had to switch primaries or not, my setup is now complete, so it's time to run the insert commands in the problem:

```
z:PRIMARY> db.foo.insert( { _id : 1 }, { writeConcern : { w : 2 } } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY> db.foo.insert( { _id : 2 }, { writeConcern : { w : 2 } } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY> db.foo.insert( { _id : 3 }, { writeConcern : { w : 2 } } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY>
```

```
z:PRIMARY> var a = connect("localhost:27001/admin");
connecting to: localhost:27001/admin
z:PRIMARY> a.shutdownServer()
2015-05-01T16:11:42.920-0400 DBClientCursor::init call() failed
server should be down...
z:PRIMARY> rs.status()
{
        "set" : "z",
        "date" : ISODate("2015-05-01T20:26:26Z"),
        "myState" : 1,
        "members" : [
                {
                        "_id" : 1,
                        "name" : "localhost:27001",
                        "health" : 0,
                        "state" : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        "optime" : Timestamp(1430511923, 3),
                        "optimeDate" : ISODate("2015-05-01T20:25:23Z"),
                        "lastHeartbeat" : ISODate("2015-05-01T20:26:24Z"),
                        "lastHeartbeatRecv" : ISODate("2015-05-01T20:26:18Z"),
                        "pingMs" : 0,
                        "syncingTo" : "localhost:27003"
                },
                {
                        "_id" : 2,
                        "name" : "localhost:27002",
                        "health" : 1,
                        "state" : 7,
                        "stateStr" : "ARBITER",
                        "uptime" : 1443,
                        "lastHeartbeat" : ISODate("2015-05-01T20:26:26Z"),
                        "lastHeartbeatRecv" : ISODate("2015-05-01T20:26:24Z"),
                        "pingMs" : 0
                },
                {
                        "_id" : 3,
                        "name" : "localhost:27003",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 1473,
                        "optime" : Timestamp(1430511923, 3),
                        "optimeDate" : ISODate("2015-05-01T20:25:23Z"),
                        "electionTime" : Timestamp(1430511789, 1),
                        "electionDate" : ISODate("2015-05-01T20:23:09Z"),
                        "self" : true
                }
        ],
        "ok" : 1
}
z:PRIMARY>
```

At this point, I've still got both my primary and my arbiter, which is enough to give my server the votes it needs to remain primary. Let's insert some more documents:

```
z:PRIMARY> db.foo.insert( { _id : 4 } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY> db.foo.insert( { _id : 5 } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY> db.foo.insert( { _id : 6 } )
WriteResult({ "nInserted" : 1 })
z:PRIMARY> db.foo.find()
{ "_id" : 1 }
{ "_id" : 2 }
{ "_id" : 3 }
{ "_id" : 4 }
{ "_id" : 5 }
{ "_id" : 6 }
z:PRIMARY>
```

I can see that I have 6 documents, so that is my answer.