Name: _____

Directions: **Work only on this sheet** (on both sides, if needed); do not turn in any supplementary sheets of paper. There is actually plenty of room for your answers, as long as you organize yourself BEFORE starting writing.

IMPORTANT NOTE: In all problems that ask you to modify an existing program, your answer should consist of statements of the form "Delete line 23," "Replace line 168 by the following code," and "Between lines 8888 and 8889 insert the following code." Also, if one part of a multipart problem asks you to modify code, the modifications apply to that part only.

1. (10) Consider the program **SMP.py**. As we run a **Processor** thread, what variable stores the ID number of the memory unit, if any, currently being used by this thread?

2. This problem concerns our example program **CallCtr.py** (the "nurse pool").

(a) (15) Add code that computes the mean lifetime of all timebombs.

(b) (15) Add code, using **Monitor.timeAverage()**, that computes and prints the average number of nurses online.

(c) (15) Give an expression, in terms of program variables, for the number of patients currently talking to a nurse.

3. Consider our example program **QoS.py**, which simulates a video/data transmission channel.

(a) (15) Suppose that we had inadvertently omitted the **return** statement on line 46. Then which one of the following would occur?

(i) The program would deadlock.

(ii) The program would halt from an execution error due to an empty event list.

(iii) The program would go into an infinite loop.

(iv) The program would run to completion, but the reported mean delay would be too small.

(v) The program would run to completion, but the reported mean delay would be too large.

(b) (15) Add code to **main()** (nowhere else) that will compute and print out the mean number of data packets in the system.

4. (15) Add a function **PropLess()** to the **Monitor** class that computes and returns the fraction of recorded observations that are less than **Cutoff**, where the latter variable is the sole argument to **PropLess()**.

Solutions:

1. self.Module

2a.

```
41.5:      self.TBStarted = None
           self.TBMon = Monitor()
51.5:      self.TBStarted = now()
        def EndTimeBomb(self):
           self.cancel(self.TB)
           self.TBMon.observe(now()-self.TBStarted)
           self.TB = None
           self.TBStarted = None
63,64:     self.EndTimeBomb()
77.5:      self.EndTimeBomb()
153.5:     print 'mean lifetime of timebombs =',G.NrsPl.TBMon.mean()
```

2b.

```
41.5:      NPtsTalking = 0
           PtsTalkingMon = Monitor()
111.5:     G.NrsPl.NPtsTalking += 1
           G.NrsPl.PtsTalkingMon.observe(G.NrsPl.NPtsTalking)
112.5:     G.NrsPl.NPtsTalking -= 1
           G.NrsPl.PtsTalkingMon.observe(G.NrsPl.NPtsTalking)
153.5:     print 'mean number of patients talking =', \
              G.NrsPl.PtsTalkingMon.timeAverage()
```

2c.

PtClass.NPtsInSystem-len(G.NrsPl.Rsrc.waitQ)

3a. The video packets which should get discarded are retained, making the queue, and queuing time, longer, so the answer is (v).

3b. Use Little's Rule:

```
MeanDataDelay = DataArrivals.TotWait/DataArrivals.NSent
print 'mean number data packets in system:', DataArrivals.DArrRate * MeanDataDelay
```

Note that it doesn't matter that the data packets are mixed in with the video packets. Little's Rule is solely an issue of flow. So, if we view the data packets in isolation, ignoring video, and still get the correct mean count.

4.

```
def PropLess(self,Cutoff):
   count = 0
   for tup in self:
      if tup[1] < Cutoff: count += 1
   return float(count)/len(self)
```

This could be made faster and more compact by using Python's **reduce()** function.