CSc 422 — Homework 3

Due Thursday, April 14, 2005 *in class*

This assignment is again worth 40 points, divided as indicated. Append a commented listing of your program to your answers to the questions. Also submit your program electronically as described at the end of the assignment.

You may as usual discuss the meanings of questions with classmates, but the work that you turn in must be yours alone. Explain your answers clearly and succinctly.

1. [8 points] MPD book, Exercise 5.14.

2. [4 points] MPD book, Exercise 5.16.

3. [4 points] MPD book, Exercise 7.3, part (a).

4. [4 points] MPD book, Exercise 7.8.

5. [20 points] Write a program to simulate the Roller Coaster problem described in Exercise 5.17, part (a). Simulate the passengers and car as processes (threads) and program a monitor to synchronize their actions.

Write your program in MPD, in Java, or in C plus Pthreads. If you use MPD, structure your program as in `rw.simulation.mpd` and simulate monitor entry, exit, wait, and signal using semaphores as shown in Figure 6.7. If you use Java, see Section 5.4 for how to program threads and synchronized methods. If you use Pthreads, see Section 5.5 for how to program monitors. There are sample programs for all three languages in `/home/cs422/Programs`.

Your program should have four command-line arguments in the following order:

> `n`, the number of passengers;
> `C`, the capacity of the coaster car;
> `numRides`, the number of rides taken by each passenger; and
> `rideTime`, the number of seconds it takes for the car to go around the track

Each passenger should repeatedly "do other things" for a random amount of time, then take a roller coaster ride. The coaster repeatedly waits until `C` passengers have requested rides, then goes around the track for `rideTime` seconds, then lets the passengers out of the car. The program should terminate after each passenger has taken `numRides` rides. It is OK for the car to be partially full on its last trip around the track.

You might want to use additional command-line arguments when developing your program. For example, you might want to seed the random number generator so your output is deterministic. However, make sure the program you turn in can run correctly with just the above four arguments.

The monitor should have three procedures as specified in Exercise 5.17:

`takeRide`, called by passengers
`load`, called by the car to get passengers
`unload`, called by the car at the end of a ride

The output of your program should be a trace of the significant events that occur: passengers calling `takeRide`, the car calling `load`, the car starting around the track (return from `load`), the car calling `unload`, and passengers getting off the car (return from `takeRide`). Each line of the trace should contain a time stamp—e.g., from the MPD `age()` function—and a short descriptive message. For example, three lines of the trace might be:

```
1203:  Passenger 6 calls takeRide()
1245:  Passenger 4 calls takeRide()
1245:  Car returns from load() and starts around track
```

Write the trace to the standard output file.

Turn in a printed copy of your program listing, and the output from two interesting (and nontrivial) runs of your program. Again follow the programming style described in the handout for Homework 1.

Also turn your program in electronically on Lectura. The assignment name is `hw3.coaster`. The file name should be `coaster.mpd`, `coaster.java`, or `coaster.c`, depending on whether you use MPD, Java, or C plus Pthreads.