# COMP3500: Project 4 - Part 4 The Cats-and-Mice Implementation

**Exercise 1:** Please complete the following turn-switch algorithm.

/\*Case 1: there are waiting mice\*/

if (there are waiting mice) {

turn\_type = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

mice\_in\_this\_turn = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

print “It is mice turn now.”;

}

else if (there are waiting cats) { /\* case 2 \*/

/\*let cats eat \*/

cats\_in\_this\_turn = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

else {/\*no waiting cats or mice\*/

turn\_type = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

}

/\* **Wake up those waiting for turn change**\*/

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

**Exercise 2:** (2.1)What items should you initialize in the driver code of catmouselock()?

(2.2) How to cleanup in catmouselock()?

**Exercise 3:** Please follow the sample code to create mouse threads using thread\_fork().

**Exercise 4:** (4.1) What is the condition under which this parent process (i.e., catmouselock()) has to wait for the cat and mouse threads?

(4.2) Can you let the parent process wait using cv\_wait()?

**Deadlocks Introduction**

**Exercise 5:** Can a deadlock occur? How?



**Exercise 6:** Space is available for allocation of 200Kbytes, and the following sequence of events occur:



**Exercise 7:** Can you explain this resource allocation graph? Is there any problem with this resource allocation graph?

