COS40003 Concurrent Programming

Programming Test:

Duration: 60 min

Problem: Meeting Professor Problem

At the end of the semester, students are queuing to ask Professor questions. At one time, only one student can consult in Professor’s office, the consultation takes a fixed **three seconds**. There are **two seats** for students to sit and wait outside Professor’s office. Suppose students come **every second**.

If Professor finds any student is waiting, he will let a student in and explain whatever for three seconds, and let the student leave. He will keep doing this. If no one is waiting, he will rest.

If a student comes and finds there is no waiting seat available, the student will leave; otherwise, the student will sit and wait. If a waiting student finds Professor is resting, the student will wake up Professor, consult and leave; otherwise the student will sit waiting to be called, and then consult and leave.

Write a concurrent program to simulate the above process. Output is like:

```
Student 1 just sat down.
Student 1 is consulting.
Professor is explaining.
Student 2 just sat down.
Student 3 just sat down.
Professor is explaining.
Student 2 is consulting.
Student 4 just sat down.
No available seat. Student 5 just left.
No available seat. Student 6 just left.
Professor is explaining.
Student 3 is consulting.
Student 7 just sat down.
No available seat. Student 8 just left.
No available seat. Student 9 just left.
Student 4 is consulting.
Professor is explaining.
……
Professor is explaining.
Student 7 is consulting.
……
```

Here, printing order in dashed rectangle is not required:

This means both

“Professor is explaining.
Student X is consulting.”

and

“Student X is consulting.
Professor is explaining.”

are fine.

Marking (20 Marks)

- correct output and the code has correct concurrency logic (15 marks)
- code inspection: clean and clear (5 marks)

Code Submission

- Please submit your code to canvas before the submission deadline.