(1) What are the four necessary conditions for a process deadlock to occur?
   - Mutually-exclusive resources
   - Non-preemptive resources
   - Hold & wait
   - Circular wait

(2) Why is it difficult to eliminate the condition of “non-preemptive resources” to prevent a process deadlock from occurring?

Some resources are non-preemptive by their natures (most of the printers, and DVD/CD-burners). Making such resources preemptive defeats their purposes.

(3) Which of the following sentence is the correct definition of “safe state” in deadlock avoidance (select the best option)?

(a) From the current state, deadlock can never happen no matter what happen after the current state.
(b) From the current state, there is at least one particular way that lets all the processes complete without causing a deadlock.
(c) The deadlock has already happened.
(d) If deadlock has not happened yet that is a safe state.
(e) None of the above

Solution: (b)
(4) Why is it difficult to eliminate the condition of “hold & wait” (by applying “request all after you drop what all you currently hold” method) to prevent a process deadlock from occurring?

This approach is difficult to use, since it causes process starvation especially to those processes that require a large number of resources.

(5) In the classroom, we discussed what we can do to make sure one of the four necessary conditions for a deadlock is not satisfied. Is it possible to have a technique that never causes “circular wait”? If yes, describe how.

One of the solutions for preventing the condition of “circular wait” is:

(1) Assign a unique resource ID to each resource available
(2) Require every process to acquire (request) resources only in the ascending order of the resource IDs