(1) What is “non-preemptive process scheduling”?

The non-preemptive process scheduling is a type of process scheduling algorithms, in which once a processor is assigned to a process, the processor will not be taken away from the process unless (a) the process finishes running or (b) the process voluntarily releases the processor (e.g., for performing I/O requests, such as waiting for inputs from the keyboard).

(2) How can “race condition” happen? Show “how” using an example.

The race condition is the situation where the outputs from processes are never predictable (i.e., each time the process is executed, its outputs can be different), because of unpredictable timing of the process to be preempted by another process.
(3) What is “critical section”?

The term “critical section” means a portion of a process where race condition possibly occurs.

(4) What does “atomic” in “atomic operations” mean?

The term, “atomic” in “atomic operations”, means the type of operations (instructions, commands, and activities) in a computer system that is always (or guaranteed to be) executed to its completion once an atomic operation starts (i.e., a processor starts executing one). For example, each machine code (or assembly instruction) is atomic, while a high-level programming language statement (e.g., “if (variable_x < 0)”) or “i = i + 1 (including “i++”) is not atomic.

(5) What “signal” system call to a semaphore exactly performs?

Signal

• If no one waiting on S, set S = 1
• If some one waiting on S, let the first proceed to CS and leave S = 0