The ThreadQuestionOne class contains a static ariable of type Random but it is not used.

Definition of the run() method:
The method prints a line that consists of two concatenated strings. The first part is the information passed to the constructor, the second is an arrow pointing away. The next line printed contains the same structure as the first but the arrow points tot the opposite direction.

If the info passed to the object is equal to “first” there is a new ThreadQuestionOne object is created with parameter “fourth”. This new object is not saved into a variable but it is made “executable”, not guaranteed to run but made available to run to the scheduler.

Line 22:
A new ThreadQuestionOne object is created saved in the variable aT4_0. The parameter info receives the value “first”. Since the run method for the object aT4_0 has not run yet, the new thread with info “fourth” is not created yet.

Line 23:
A new ThreadQuestionOne object is created saved in the variable aT4_1. The parameter info receives the value “second”.

Line 24:
A new ThreadQuestionOne object is created saved in the variable aT4_2. The parameter info receives the value “third”.

Line 26:
At this moment the scheduler does not have any threads to execute. For the variable aT4_0, because the run() method is called, if there was anything made available for execution, aT4_0 takes precedence. At the end of the the thread aT4_0, ThreadQuestionOne with info “four” is made available to execute, and scheduled but the scheduler.

Line 27:
Currently the scheduler is either running aT4_0 or there is the ThreadQuestionOne with info “four” is made executable and it’s waiting. Using the run() method on aT4_1, the thread is run immediately skipping the “executable” state meaning the scheduled thread in aT4_0 is not run immediately.

Line 28:
At this point either aT4_1 is running or it ended, and in this line aT4_2 is being scheduled. However, since the thread object scheduled in the thread with info “first” is still waiting for execution as soon as aT4_1 ends, that thread will begin. Finally, aT4_2 will be executed once all are complete. Then the program will end.

Is the following output possible?

first --->
fourth --->
first <---
second --->
fourth <---
third --->
second ←-
third <--

It is not possible to get this output as the thread with info “fourth is scheduled after the print statements of the first thread is executed.