. (a) The program has a bug which you should correct. Run the program through JIVE and observe the object diagram in order to understand the cause of the bug. Name your corrected file as Readers_Writers_Corrected.java. Run it through JIVE and make sure that the program is behaving correctly. Provide an explanation of the bug in a file explain.pdf.

Ans: The bug is that instead of using notifyAll() , the code was using notify(). Now the difference that comes along with in mainly 2 conditions, in the 2 boundary conditions:

a) when the database is full then suppose a writer is in the critical section then it should wait and because no further writes are possible. However, in this case a read should happen and suppose a reader is waiting to read on the resource, in that case both reader and writer are waiting on the resource. Now if a notify is called and in that case what is going to happen is that it can either give a reader or writer the access if in case the access is given to the writer then we go into a deadlock as the reader was also waiting on the shared resource. Hence notifyAll() will wake all the waiting queues, which solves our problem.

b) The same happens when the database is empty, if both the readers and writers are trying to get a lock over the shared resource in that case since thread scheduler randomly wakes either the reader thread or writer thread in that case, if a reader thread is woken up and not a writer thread which just kept on waiting will give us a deadlock state.