## Activity 5 - Learning by doing 2

Stack stack = new Stack();

while (!q.isEmpty())

stack.push(q.dequeue());

while (!stack.isEMpty())

q.enqueue(stack.pop());

In the above code fragment, elements from the queue are dequeued/popped and inserts into stack. So, the new stack will get the all the elements of queue in the reverse order of queue.

Again, pops each element of stack and insert back into the queue. Then resultant queue will get the same elements as first, but in reverse order.It is because of the FIFO of queues.

Example Trace: Queue : edcba and Stack : Empty

|  |  |
| --- | --- |
| **Queue** | **Stack** |
| edcba |  |
| edcb | a |
| edc | ab |
| ed | abc |
| e | abcd |
|  | abcde |

|  |  |
| --- | --- |
| **Stack** | **Queue** |
| abcde |  |
| abcd | e |
| abc | de |
| ab | cde |
| a | bcde |
|  | abcde |