The code starts off like this:

```java
if (head != null){
    headList1 = null;
    headList2 = head;
}
```

That’s just saying that if the list is not empty, then there is something to do (namely reverse the list). In that case, we’re going to initialize headList1 and headList2 as above.

That gives us the situation shown on the next slide.
headList1 → null

head
headList2
Then, supposed we run the following four instructions. I claim that this will transfer one node from list2 to list1.

```
cur = headList2.next;
headList2.next = headList1;
headList1 = headList2;
headList2 = cur;
```
cur = headList2.next;
headList2.next = headList1;
headList1 = headList2;
headList2 = cur;
cur = headList2.next;
headList2.next = headList1;
headList1 = headList2;
headList2 = cur;
cur = headList2.next;
headList2.next = headList1;
headList1 = headList2;
headList2 = cur;
cur = headList2.next;
headList2.next = headList1;
headList1 = headList2;
headList2 = cur;

After the four instructions, we have one more node in the reversed part of the list (list1), and one fewer node in the not-yet-reversed part of the list (list2).
These four instructions are repeated over and over again in a while loop, until eventually head2List2 is null and headList1 references the last node in the original list.