| Operations for ArrayList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 1,000 | 0.02917 |
| Sort | 1,000 | 4.788 |
| Shuffle | 1,000 | 0.728 |
| Random Get | 1,000 | 45.118 |
| Sequential Get | 1,000 | 0.09211 |

| Operations for LinkedList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 1,000 | 0.01241 |
| Sort | 1,000 | 3.625 |
| Shuffle | 1,000 | 0.609 |
| Random Get | 1,000 | 839.66 |
| Sequential Get | 1,000 | 0.96623 |

| Operations for ArrayList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 5,000 | 0.02117 |
| Sort | 5,000 | 24.55 |
| Shuffle | 5,000 | 2.864 |
| Random Get | 5,000 | 85.624 |
| Sequential Get | 5,000 | 0.75092 |

| Operations for LinkedList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 5,000 | 0.00541 |
| Sort | 5,000 | 9.726 |
| Shuffle | 5,000 | 2.915 |
| Random Get | 5,000 | 5474.951 |
| Sequential Get | 5,000 | 27.9876 |

| Operations for ArrayList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 10,000 | 0.0074 |
| Sort | 10,000 | 25.672 |
| Shuffle | 10,000 | 5.055 |
| Random Get | 10,000 | 69.526 |
| Sequential Get | 10,000 | 1.62255 |

| Operations for LinkedList | Number of Objects | Average Time of 10 Runs (in milliseconds) |
| --- | --- | --- |
| Add | 10,000 | 0.00542 |
| Sort | 10,000 | 17.058 |
| Shuffle | 10,000 | 8.554 |
| Random Get | 10,000 | 18092.493 |
| Sequential Get | 10,000 | 165.26583 |

The Add, Sort, and Shuffle operations are faster when implemented by a LinkedList when there are 1,000 objects. However, based on the average runtimes, it seems that ArrayLists shuffle objects more quickly when there are 5,000 and then 10,000 objects. It is also apparent that random gets and sequential gets are faster when implemented by ArrayLists. The results are consistent with how the list is implemented because the ArrayList is faster when it comes to accessing elements, but slower when it comes to adding elements. The LinkedList is faster when it comes to adding elements, but slower when it comes to accessing elements.