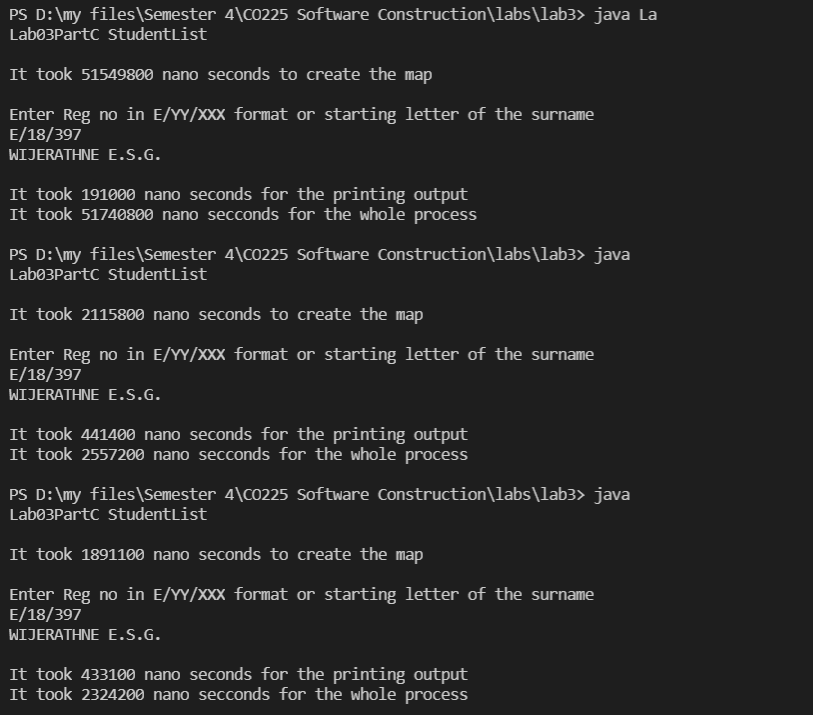
**CO 225 - Lab 03 – Part C**

**Wijerathne E.S.G. – E/18/397**

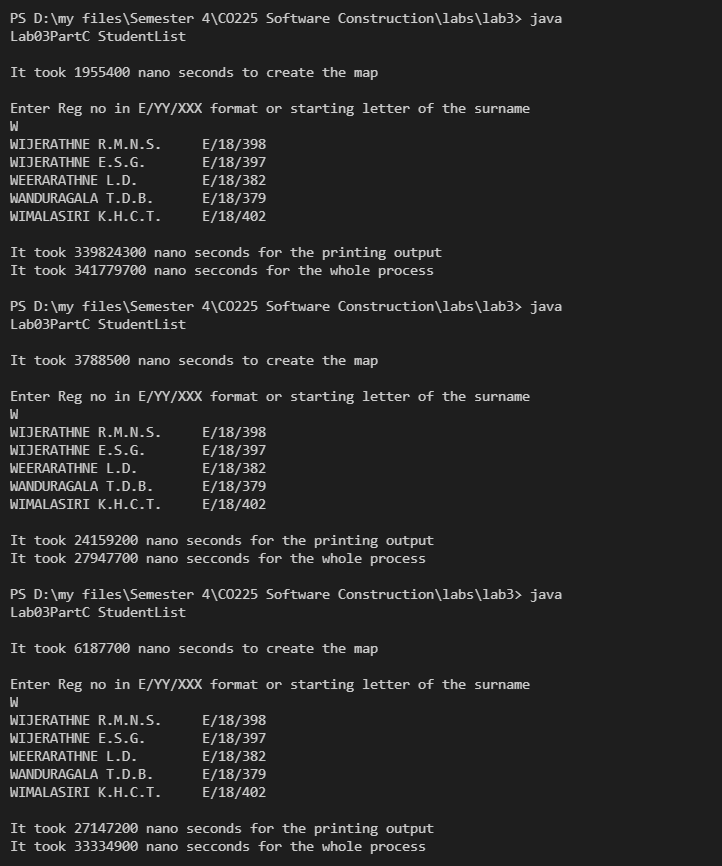
* Lab03PartC.java was implemented by using Map
* Lab03PartD.java was implemented by using List
* Lab03PartE.java was implemented by using Set

**Running Times**

Lab03PartC – When input is E No.



Lab03PartC – When input is letter.

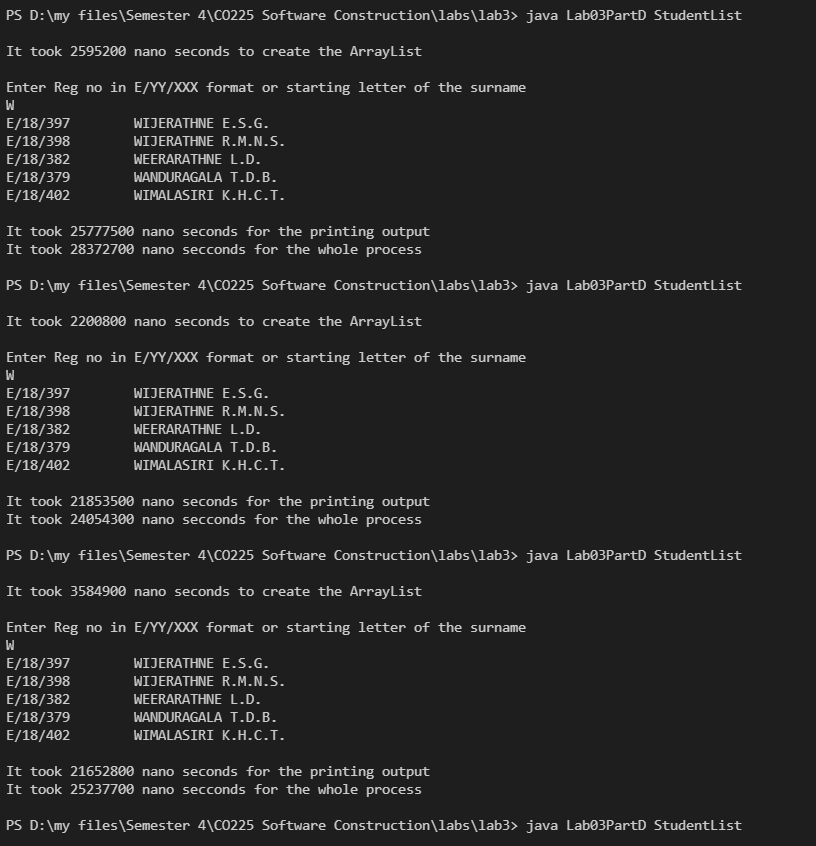


* A Map is an object that maps keys to values. A map cannot contain duplicate keys: Each key can map to at most one value. It models the mathematical function abstraction.

Lab03PartD – When input is E No.

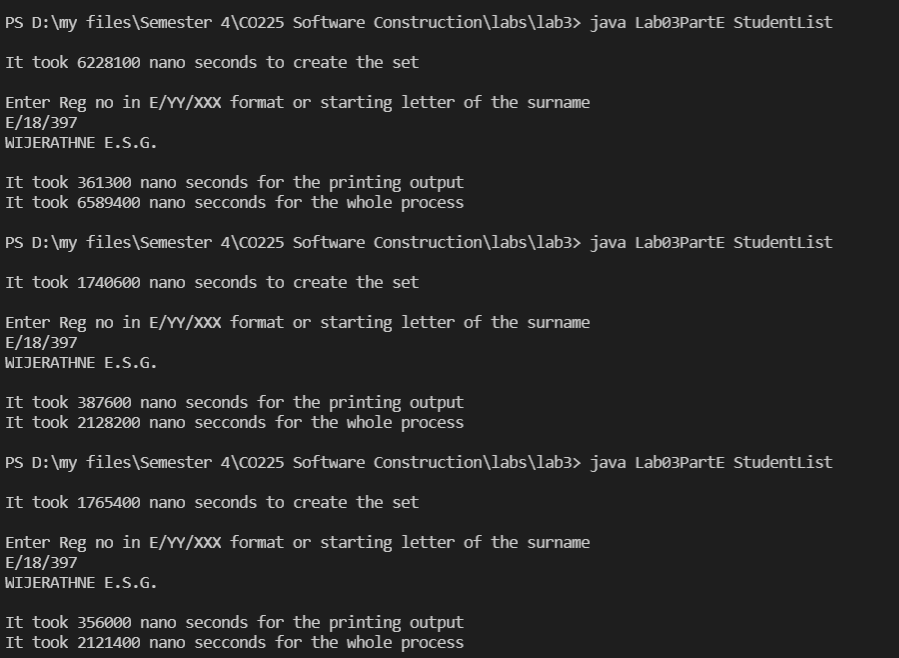


Lab03PartD – When input is letter.

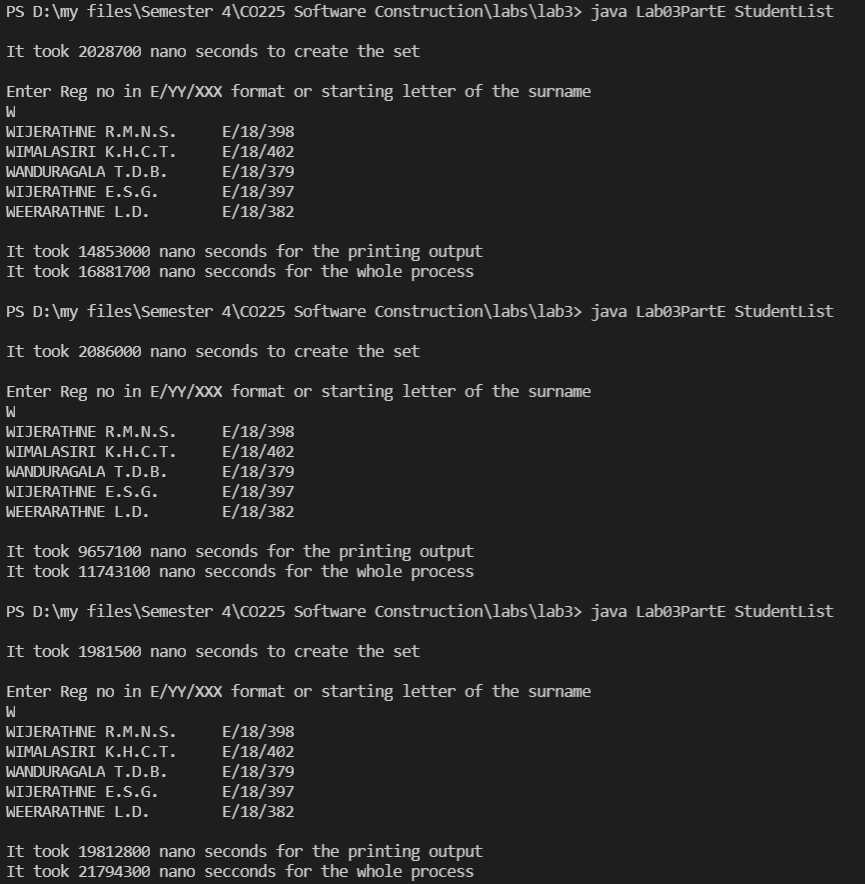


* List in Java provides the facility to maintain the ordered collection. It contains the index-based methods to insert, update, delete and search the elements. It can have the duplicate elements also.

Lab03PartE – When input is E No.



Lab03PartE – When input is letter.



* The set is an interface available in the java.util package. The set interface extends the Collection interface. An unordered collection or list in which duplicates are not allowed is referred to as a collection interface. The set interface is used to create the mathematical set. The set interface use collection interface's methods to avoid the insertion of the same elements.

**Conclusion**

Calculating Average Times using above results

|  |  |  |  |
| --- | --- | --- | --- |
| **Trial** | **making time in nanoseconds** | | |
| **Map** | **List** | **Set** |
| 1 | 2115800 | 2176500 | 6228100 |
| 2 | 1891100 | 2611900 | 1740600 |
| 3 | 3339800 | 3623200 | 1765400 |
| 4 | 3788500 | 2200800 | 2028700 |
| 5 | 6187700 | 3584900 | 2086000 |
| 6 | 4388300 | 3238200 | 1981500 |
| **Average time** | **3618533** | **2905917** | **2638383** |

Table 01: Time to insert data into different data instructions

* As seen in above table inserting data into set is much faster.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Trial** | **Map** | | **List** | | **Set** | |
| **E.No as inp.** | **Letter as inp.** | **E.No as inp.** | **Letter as inp.** | **E.No as inp.** | **Letter as inp.** |
| 1 | 441400 | 24159200 | 254700 | 21853500 | 361300 | 14853000 |
| 2 | 433100 | 27147200 | 439500 | 21652800 | 387600 | 9657100 |
| 3 | 472700 | 29529800 | 485000 | 25390300 | 356000 | 19812800 |
| **Average time** | **449066.67** | **26945400** | **393066.67** | **22965533.3** | **368300** | **14774300** |

Table 02: Time take by two operations in different data structures (Time is in nano seconds)

* As seen in this table implementations of lab 03-part C functionalities also fast in set implementation