

**CO225: Software Construction**  
**Lab 03: Arrays and Collections**

**E/18/128**

D . Compare the ease of implementation, and the performance of the above three different Java programs (Lab03PartC, Lab03PartD, Lab03PartE).

**Lab03PartC**

HashMap has been used for this part. It is like a dictionary in python. For this implementation there is no need to build classes and include those objects like 2 other implementations. It can directly add register Number and name as key and value respectively. It does not allow duplicate keys therefore it will be made sure that there are no students with the same register number. There are many methods in hashmap. We can use .containsKey() to check whether a given register number is present and also methods like .getKey() .getValue() are available. Comparatively faster than other implementations. Hashmap get(key) can be  $O(1)$  in the best case and  $O(n)$  in worst case time complexity

**Lab03PartD**

ArrayList has been used for this part. It implements List interface. When we need the order of student details same as in the file "StudentList", Arraylist is most suitable since HashSet and HashMap don't maintain the insertion order. Duplicates may be stored in ArrayList. get(index) method always gives  $O(1)$  time complexity

**Lab03PartE**

HashSet has been used. It implements Set interface. Duplicates are not allowed here like hashmap but to get that performance we have to override the equals() and hashCode() methods. It internally uses HahTable for its implementation. Time complexity for HashSet .add is  $O(1)$ .