Hello everyone.

This week we’re asked to discuss how we would choose a data structure from the Collection Framework to use in a program.

This is a interesting topic for me since it what it really boils down to is, what type of program you’re trying to code. Therefore, I will list all the necessary steps to follow when choosing the right data structure to use in a program. All listed ideas, are ideas that I think is best to consider when choosing, therefore, none of the ideas are facts, but, rather options to keep in mind that are all related to my opinion regarding this subject.

There are quite a few data structures that are listed in the Collections Framework. However, they all fall under certain categories. Lists, Sets, and Maps are the three main categories, however, in side those categories we have multiple sub categories (Eck, 201 8). However, in this discussion I will only bother with the three main categories. Therefore, when choosing a data structure to be used in your program, you have to ask yourself, what is this program supposed to do, is efficiency more important or is it more important to have the data sorted in order, is there something that will point directly to the data in data structure, or will there only be one element representing the data? Furthermore, after asking yourself a couple of question, you can start to solve the problem and choose the right data structure for your program. For example, I want to write a program that will keep track of people based on their names, however, I also want add their phone number to their name, I would also like this program to represent the names in alphabetical order for better readability. Therefore, I can already see which data structure will be the best for my program. I will go with a TreeMap. By using a TreeMap, I might lose some efficiency, however, I will get the result I want. My data will be sorted in alphabetical order and I can match each persons name with their phone number.

In conclusion, it’s quite easy to choose a data structure to use, you just need to ask yourself a couple of logical question and you can make the process a lot easier.

**Words 381**

**References**

Eck, J, D. (2018). *Introduction to Programming Using Java.* Hobart and William Smith Colleges