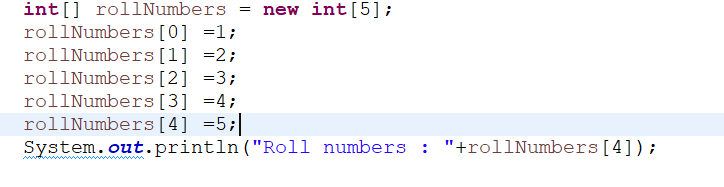
Collections

Why collections?

Problem statement: Need to store 100 roll numbers of students.

For the above scenario, we need to create 100 variables to store the values.

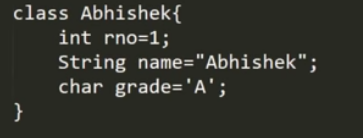
To avoid this, we have arrays which stores all these roll number values in a single variable.



But if we want to add names of the students along with roll numbers, arrays will not support anymore, as the arrays are limited for storing similar data types.

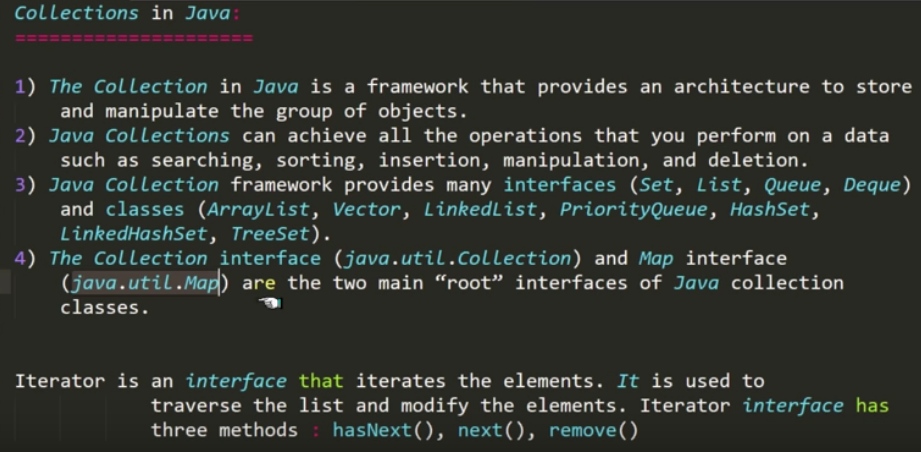
One other disadvantage of using arrays is the size. Size is fixed and can’t be changed.

To store different data types we can have an object created for a class (we will store different datatypes here).



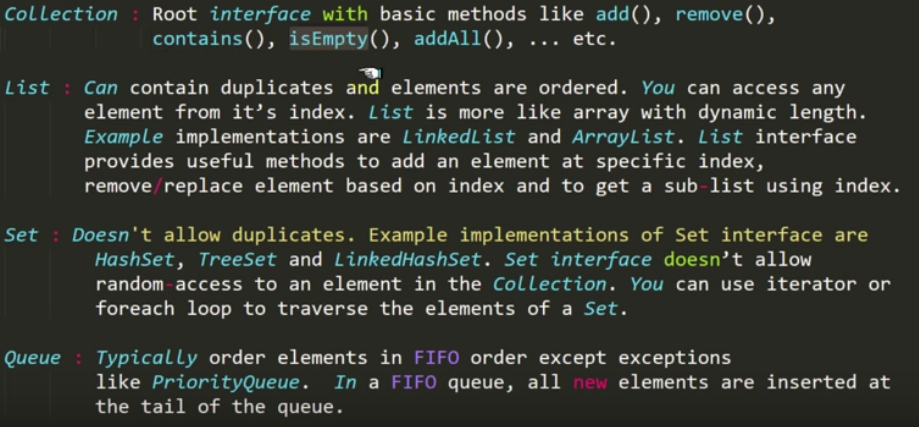
But changing or adding these values at runtime is another difficulty here.

To overcome these problems, collections are introduced.



Collections – framework

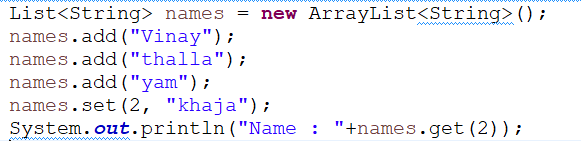
Collection - Interface

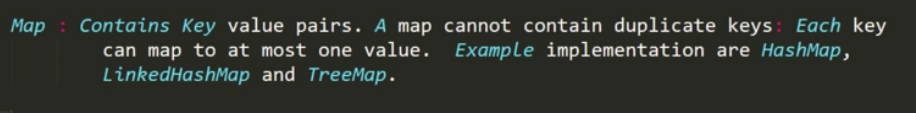


\*\*\* Objects can’t be created for interfaces. So we have to implement this interface by any class then we can create an object for this class and make use of the methods that are present.

Subclasses which will implement this interfaces can be directly used by creating objects to them.

Ex:





**Summary**

