Need of Collections:

An Array is an indexed collection of fixed number of homogeneous data elements.

The main advantage of Arrays is we can represent multiple values with a single variable.

So that reusability of code will be improved.

Limitations of Arrays:

1. Arrays are fixed in size i.e., once we created an array with some size there is no chance of increasing or decreasing it’s size based on our requirement. Hence to use arrays compulsory we should know the size in advance which may not possible always.
2. Arrays can hold only homogeneous data elements:

Ex: Student s[]=new Student[100];

S[0]=new Student();//correct

S[1]=new Customer(); //wrong

But we can resolve this problem by using Object arrays.

Object o[]=new Object[100];

o[0]=new Student();//correct

o[1]=new Customer(); //correct

Arrays concept is not implemented based on some standard data structure hence readymade method support is not available for every requirement we have to write the code explicitly. Which increases the complexity of the program.

**To overcome the above limitations of Arrays we should go for collections.**

Collections are growable in nature.i.e. Based on our requirement we can increase (or) decrease the size.

Collections can hold both homogeneous & heterogeneous elements.

Every collection class is implemented based on some standard data structures. Hence readymade method support is available for every requirement. Being a programmer we have to use this method and we are not responsible to provide implementation.

**Differences between Arrays and Collections:**

**1)Arrays are fixed in size.**

**1)Collections are Growable in size.**

**2)with respect to memory Arrays are not recommended to use.**

**2) with respect to memory collections are recommended to use.**

**3) with respect to performance wise Arrays are too good.**

**3) with respect to performance wise collections are not recommended.**

**4) Arrays can hold only homogeneous elements.**

**4)collections can hold both can hold Homogeneous & heterogeneous elements**

**5)There is no underlying data structure so no ready made method are not available.**

**5) Every collection class is implemented based on some standard data structure so that ready made support is available.**

**6) we can create int[] and Integer[]. (primitives and Objects can be hold.**

**6)we can hold Objects only. Not primitives.**

|  |  |
| --- | --- |
| Arrays | Collections |
| 1. **Arrays are fixed in size.** | **1.Collections are Growable in nature. i.e., based on our requirement we can increase or decrease the size.** |
| 2.With respect to memory arrays are not recommended to use. | **2.with respect to memory collections are recommended to use.** |
| 3.With **respect to performance Arrays are** recommended to use. | **3.with respect to performance collections are not recommended to use.** |
| **4.Arrays can hold only homogeneous data type elements.** | **4.Collections can hold both homogeneous & heterogeneous elements.** |
| **5.There is no underlying data structure for arrays and hence ready made method support is not available.** | **5.Every collection class is implemented based on some standard data structure. Hence ready made method support is available** For every requirement. |
| **6.we can hold both int[] and Integer[]. (primitives and Objects types).** | 1. **Collections can hold only Objects but not primitives.** |