How the values of ArrayList are stored in the memory?

- In ArrayList the values that are stored in it, are boxed into the object type. Then, we have to unbox them.
- Because of boxing the objects, we can store the objects of different types.
- If you store only objects of reference types in ArrayList , then boxing is not used.
- In the ArrayList , boxed values are stored in the heap memory and unboxed values are stored in the stack memory.

==========

==== What are the disadvantages of ArrayList (Collection – ArrayList)?

1. Every time we have to unbox, to get the values.
2. If there is a chance of assigning a wrong datatype then we may get runtime errors. Runtime errors are dangerous than Compile time errors.

==========

Write the differences between Collections and Generics?

	Collections	Generics
namespace	System. Collections;	System .Collections .Generics;
Each element is of what type	Object	It can store any elements based on input.
Do you need Type casting	Yes, we need Typecasting because the values are boxed.	No need of Typecasting because it stores same kind of values.

Example:	ArrayList data=new ArrayList();	List<int> data=new List<int>();

How the values of List<T> are stored in memory?

- Generally, List<T> stores objects of same type.
- It implements IList generic interface using array whose size is dynamically increased as required.
- The values are stored in the managed Heap.
- In List<T> makes a single array of <Type>, and can store the values directly. If List<int> means it makes a single array of integers and store the values directly.

Data Types and their respective alias names:

DATATYPE	ALIAS NAME
1. byte	Byte
2. ushort	UInt16
3. uint	UInt32
4. ulong	UInt64
5. sbyte	SByte
6. short	Int16
7. int	Int32
8. long	Int64
9. float	Single
10.double	Double
11.decimal	Decimal

12.char	Char
13.String	String
14.bool	Boolean