1 . **How the values of array list are stored in memory:**

* The elements of an Array List are stored in a chunk of contiguous memory.
* That memory becomes full, a larger chunk of contiguous memory has to be allocated which is usually twice the size and the existing elements are copied into this new chunk.
* We call this chunk the capacity of the Array List object.

2. **Disadvantage of arraylist(collection array list) :**

* Dynamically size is increased.
* Each value in a object have to be unbox.
* There is a chance of giving wrong datatype we can get run time errors.

**3 .differences between Collections and generics :**

|  |  |  |
| --- | --- | --- |
|  | collections | generics |
| namespace | System.collections | System.collection.genric |
| Element type | object | int |
| Type casting | yes | no |
| Example: | ArrayList data =New ArrayList(); | List<int>data =New List<int>(); |

**4. Datatypes alias names:**

|  |  |
| --- | --- |
| datatype | Alias datatype |
| byte | System. Byte |
| ushort | System.UInt16 |
| uint | System.UInt32 |
| ulong | System.UIN64 |
| sbyte | System. SByte |
| short | System.Int16 |
| int | System. Int32 |
| long | System.Int64 |
| float | System. Single |
| double | System. Double |
| decimal | System. Decimal |
| char | System. Char |
| boolean | System. Boolean |
| string | System. String |

**5 . Example programs for implicit and explicit type casting:**

|  |  |
| --- | --- |
| Implict type casting | Explicit type casting |
| Short a =25;  Int b=a;  Console.Readline(); | Int a =25;  Short b=(short)a;  Console.ReadLine(); |

**6 . how the values of List<T> are stored in the memory:**

* the memory to store the value types is within the memory allocated for the System.
* ArrayList each element is just a reference to a boxed value type.
* the actual memory to store each value type is elsewhere on the Heap memory.