Day 6 morning assignment

|  |
| --- |
| **1.Difference between Collections and Generic** |

|  |  |  |
| --- | --- | --- |
|  | Collections | Generic |
| 1.Namespace | Systems.collections | Systems.collections.generic |
| 2. Each element of what type | Object | Int( any datatype it can store) |
| 3. Do we need type cast here | Yes | No |
|  | **Arraylist** | **List<T>** |
| Example | ArrayList data =New ArrayList(); | List<int> data =New List<int>(); |

|  |
| --- |
| **2.Data types in C# and their respective names** |

|  |  |
| --- | --- |
| **Data type name** | **Data type alias name** |
| byte | Byte |
| ushort | UInt16 |
| uint | UInt32 |
| ulong | UInt64 |
| sbyte | SByte |
| short | Int16 |
| int | Int32 |
| long | Int64 |
| float | Single |
| double | Double |
| decimal | Decimal |
| Boolean | Boolean |
| char | Char |
| string | String |

|  |
| --- |
| **3.Disadvantages of Arraylist (collections Arraylist)** |

1.Runtime Error

2.Each value in object have to unbox

|  |
| --- |
| **4**.**Examples programs for implicit and explicit type castings** |

|  |  |
| --- | --- |
| **Implicit type casting** | **Explicit type casting** |
| short a = 25;  int b = a;  Console.ReadLine(); | int a = 25;  short b = (short)a;  Console.ReadLine(); |

|  |
| --- |
| **5. How values of arraylist stored in memory** |

1. Arraylist changes memory allocation as it grows.
2. We specify the capacity while intializing the array list
3. It allocates enough memory to store objects upto the Capacity

|  |
| --- |
| **6.How values of List<T> stored in memory** |

1. In an ArrayList each element is just a reference to a boxed value type, so the actual memory to store each value type is elsewhere on "The Heap", i.e. somewhere "Over There"
2. Lists then ar stored in distinct chunk of memory which are linked together with pointers