Day 6 Morning Assignment

By

B.P.N.V.S.Sudheer

31-01-2022

|  |
| --- |
| 1.Create a Simple Program to declare arraylist and assign some values and find sum |
| Code : |
| using System;  using System.Collections;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace To\_declare\_array\_list  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//      static void Main(string[] args)  {  ArrayList data = new ArrayList();  int sum = 0;  data.Add(5);  data.Add(6);  data.Add(8);  foreach (var d in data)  {  sum = sum + (int) d;      }  Console.WriteLine(sum);  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| 2.Research and find how the the values of arraylist are stored in the memory |
| * The elements of an arrayList are stored in a chunk of contiguous memory.when that memory becomes full, a larger chunk of contiguous memory has to be allocated (usually twice the size)and the existing elements are copied into this new chunk.we call this chunk the capacity of the arraylist object |

|  |
| --- |
| 3.What are the disadvantages of ArrayList are stored in (collections Arraylist) |
| * And run time error will appear there is a chance of giving wrong datatype * Each value in object have to be unbox |

|  |
| --- |
| 4. Create a simple program to declare List <int> and assign some values and find sum |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace declare\_list\_int  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  static void Main(string[] args)  {  List<int> data = new List<int>();  int sum = 0;  data.Add(4);  data.Add(5);  data.Add(6);  foreach (var d in data)  {  sum = sum + d;  }  Console.WriteLine(sum);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |  |
| --- | --- |
| 5.In a tabular form write all datatypes in c# write the respectively alias name | |
| Data type name | Alias data type name |
| 1. byte 2. ushort 3. uint 4. ulong 5. sbyte 6. short 7. int 8. long 9. float 10. double 11. decimal 12. bool 13. char   14.string | Byte  ushort  UINT  ulong  SByte  Short  INT  long  single  Double  Decimal  Boolean  Char  String |

|  |
| --- |
| 6.Find how the values of list<T> are stored in the memory |
| * In a list<T>, the memory to store the value types is within the memory allocated for the system array that is over here in a 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”that is somewhere over there |

|  |
| --- |
| 7. write a c# to declare List <strings> and add 5 values and print the values using (a)for loop (b) for eachloop  (c) Lambda Expression |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace declare\_list\_string  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  static void Main(string[] args)  {  List<string> data = new List<string>();  data.Add("pavan");  data.Add("naga");  data.Add("venkata");  for (int i = 0; i < data.Count; i++)  {  Console.WriteLine(data[i]);  }  foreach (var d in data)  {  Console.WriteLine(d);  }  data.ForEach (p=> Console. WriteLine(p));  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| 8.WACP to declare list <int> and read 5 values from user and find sum using (a) for loop (b) for eachloop  (c)Lambda Expression |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace list\_int\_using\_loops  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//    static void Main(string[] args)  {  List <int> data = new List<int>();  int temp;  int sum1 = 0;  int sum2 = 0;  int sum3 = 0;  for (int i = 1; i <= 5; i++)  {  Console.WriteLine("enter any value");  temp = Convert.ToInt32(Console.ReadLine());  data.Add(temp);  }  for (int i = 1; i < data.Count; i++)  sum1 = sum1 + data[i];  foreach (var d in data)  sum2 = sum2 + d;  data.ForEach(d => sum3 = sum3 + d);  Console.WriteLine(sum1);  Console.WriteLine(sum2);  Console.WriteLine(sum3);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |  |  |
| --- | --- | --- |
| 9.In a tabular format write the difference between collections and generic | | |
| S.N.O | collections | generic |
| 1. Namespace 2. Element type 3. Type casting   Example: | ArrayList is presented in system.collections  Object type  Required  ArrayList | System.collections is presented in system.collection generic  Int type,string type  Not required  List <string>,List<int> |

|  |
| --- |
| 10. Write Example programs for implicit and explicit |
| Implicit Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace @implicit  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  static void Main(string[] args)  {  short p = 5;  int q = p;  Console.WriteLine(q);  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |  |
| --- | --- |
| Explicit code: | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace @explicit  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//  static void Main(string[] args)  {  int p = 10;  short q = (short)p;  Console.WriteLine(q);  Console.ReadLine();  }  }  } | |
| Output: | |
|  |  |