|  |
| --- |
| Day-6 Morning Assignment  By  U.Joshna  [31-1-2022] |

|  |
| --- |
| program-1 |
| Create a simple program to declare ArrayList and assign some Values and find sum. |
| Code: |
| using System;  using System.Collections;  namespace Day6\_Mrng\_Assignment\_ArrayList  {  internal class Program  {  static void Main(string[] args)  {  int sum = 0;  ArrayList data = new ArrayList();  data.Add(10);  data.Add(9);  data.Add(8);  data.Add(7);  data.Add(6);  {  sum = sum + (int)d;  }  Console.WriteLine(sum);  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |

2.Research and find how 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

.and the existing elements are copied into this new chunk

.We call this chunk the capacity of the ArrayList object

3.What are the Dis-advantages of ArrayList(Collections ArrayList)?

.The non-generic collection classes such as ArrayList,Stack,Queue,Hashtable,etc operate on the object data type

.If there is a Choice of assigning Wrong value we get runtime error

.Every time we unbox and do operation

|  |
| --- |
| Program-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 Day6\_Mrng\_Assignment\_\_List  {  internal class Program  {  static void Main(string[] args)  {  int sum = 0;  List<int> data = new List<int>();  data.Add(1);  data.Add(2);  data.Add(3);  data.Add(4);  foreach (var d in data)  {  sum = sum + d;  Console.WriteLine(sum);  }  Console.ReadLine();    }  }  } |
| Output: |
|  |
|  |

**5.**In a tabular format write the differences between Collections and Generics?

1.namespace

2.Each element is of what type

3.do you need type casting here

4.Example – ArrayList, List<T>



6.Research and 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(i.e,"Over Here").

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".

|  |
| --- |
| Program-8 |
| WACP to declare List<String> and read 5 values from user and find sum using  a.for loop  b.foreach loop  c.Lambda Expression |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_3\_Loops\_String  {  internal class Program  {  static void Main(string[] args)  {  List<string> data = new List<string>();  data.Add("Lakshna");  data.Add("Renuka");  data.Add("Kushal");  data.Add("Joshna");    //print values using for loop  for(int i=0; i<data.Count; i++)  {  Console.WriteLine(data[i]);  }    //print values using foreach loop  foreach(var d in data)  {  Console.WriteLine(d);  }  //print values using lambda expressions  data.ForEach(x => Console.WriteLine(x));  Console.ReadLine();    }  }  } |
| Output: |
|  |
|  |

8.In a Tabular format write aa data types in C# and Write the respective alias name

|  |  |  |
| --- | --- | --- |
| SNO | Data Type | 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 | Bool | Boolean |
| 13 | Char | Char |
| 14 | String | String |

|  |
| --- |
| Program-9 |
| WACP to declare List<int> and read 5 values and print the values using?  a.for Loop  b.foreach Loop  c.lambda expression |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_3\_Loops\_int  {  internal class Program  {  static void Main(string[] args)  {  List<int> data = new List<int>();  int temp;  int sum1 = 0, sum2 = 0,sum3 = 0;  //Read 3 Num from User  for (int i = 1; i <= 3; i++)  {  Console.WriteLine("enter any value :");  temp = Convert.ToInt32(Console.ReadLine());  data.Add(temp);  }  //find sum using for loop  for(int i=0; i < data.Count; i++)  sum1 = sum1 + data[i];  //find sum using foreach loop  foreach (var d in data)  sum2 = sum2 + d;  //find sum using lambda expression  data.ForEach(d => sum3 = sum3 + d);  Console.WriteLine(sum1);  Console.WriteLine(sum2);  Console.WriteLine(sum3);  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
|  |

10.Write example programs for implicit and explicit type casting?

.implicit casting(automatically) – converting a smaller type to a larger type size

Char -> int -> long -> float -> double

Example:

|  |
| --- |
| 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)  {  int myint = 7;  double mydouble = myint; //Auotomatic casting: int to double  Console.WriteLine(myint); //Output 7  Console.WriteLine(mydouble);  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
|  |

.Explicit Casting - converting a larger type to a smaller size type

double -> float -> long -> int -> char

example:

|  |
| --- |
| 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)  {  double mydouble = 3;  int myint = (int)mydouble; // explicit casting: double to int  Console.WriteLine(mydouble); //Output 3  Console.WriteLine(myint);  Console.ReadLine();  }  }  } |
| Output: |
|  |
|  |
|  |
|  |